

THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES

Etude comparative de deux méthodes de classification symbolique : DIV et DYCLASS

Laurent, Etienne

Award date:
2002

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

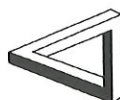
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



NDP
ulté des Sciences
artement de Mathématique

mpart de la Vierge, 8
000 Namur Belgique

Etude comparative de deux méthodes de classification symbolique : DIV et DYCLASS



Mémoire présenté pour l'obtention
du grade de
Licencié en Sciences Mathématiques
par

Promoteur : A. Hardy

Etienne LAURENT

Année académique 2001-2002

Je tiens à remercier en premier lieu Monsieur André Hardy, promoteur de ce mémoire, pour son aide et sa gentillesse tout au long de ce travail.

Mes plus sincères remerciements vont à Monsieur Yves Lechevallier, chercheur à l'INRIA, pour les nombreuses explications et sa disponibilité lors de ses déplacements à Namur.

Je tiens également à remercier, Mademoiselle Pascale Lallemand, chercheuse au département de mathématiques, pour son aide, son sourire et sa bonne humeur.

Mes pensées se tournent également vers ma famille, mes amis et mes condisciples pour leurs encouragements.

Enfin, je voudrais particulièrement remercier Mademoiselle Caroline Copin pour son aide, sa patience et son soutien quotidien.

RÉSUMÉ

Le but de la classification est de décomposer un ensemble donné de n objets, décrits par un ensemble de p caractéristiques, en un nombre relativement réduit de classes d'objets semblables.

Dans ce travail, nous étudierons et comparerons deux méthodes de classification : la méthode divisive hiérarchique de Chavent et la méthode des nuées dynamiques de Diday adaptée aux données symboliques par Lechevallier. Cette approche se fera dans le cadre de données symboliques spécifiques : les données intervalles.

Enfin, nous appliquerons ces méthodes à des jeux de données et essaierons d'en tirer quelques conclusions.

ABSTRACT

The aim of classification is to decompose a given set of n objects described by a set of p features in a relatively restricted number of clusters of similar objects.

In this paper, we will study and compare two methods of clustering : the Chavent's divisive hierarchical method and the Diday's dynamical clustering method adapted to symbolic data by Lechevallier. This approach will be done for specific symbolic data : interval variables.

Finally, we will apply this methods to datasets to try to draw some conclusions.

TABLE DES MATIÈRES

Introduction	8
Partie I. Bases de l'analyse de données	10
1. Données classiques	11
1.1. Individus et variables	11
1.2. Types de variables	12
1.2.1. Variables quantitatives	12
1.2.2. Variables qualitatives	12
1.3. Choix de la similarité	13
1.4. Matrice de données	13
2. Introduction aux données symboliques	15
2.1. Données multivaluées	16
2.2. Variables intervalles	16
2.3. Matrice des données symboliques	17
3. Notion d'inertie	18
3.1. Inertie du nuage par rapport à un point	18
3.2. Théorème de Huygens	19
3.3. Inertie associée à une partition	19
3.3.1. Les trois types d'inerties	20

3.3.2. Relation entre les inerties partielles	21
3.3.3. Décomposition des inerties sur les classes et les variables	21
4. Similarités et dissimilarités	24
4.1. Indices de similarité et de dissimilarité	24
4.1.1. Mesure de ressemblance	24
4.1.2. Indice de similarité	25
4.1.3. Indice de dissimilarité	25
4.1.4. Transformation d'un indice de similarité en indice de dissimilarité et vice-versa	26
4.1.5. Semi-ordre généré par une mesure de ressemblance	26
4.2. Indices de dissimilarité	27
4.2.1. Matrice de dissimilarité	27
4.2.2. Propriétés de différents indices de dissimilarité	27
4.3. Mesures de distance	28
4.3.1. Indices de dissimilarité pour les individus	29
Partie II. Principes de classification	32
5. Introduction à la classification	33
5.1. Problème de classification	33
5.2. Principales structures de classification	34
5.2.1. Partitions	34
5.2.2. Recouvrements	35
5.2.3. Hiérarchies	36
5.2.4. Pyramides	37
5.2.5. Formalisation de la notion d'espace de classification	38
6. Classification et données intervalles	41
6.1. Opérateur d'union jointe pour les variables intervalles	41
6.2. Mesure de Hausdorff et fonction de comparaison de Ichino	42
6.2.1. Mesure de Hausdorff	42

6.2.2. Fonction de comparaison de Ichino	44
Partie III. Méthodes de classification	46
7. Méthode hiérarchique divisive (DIV)	47
7.1. Généralités sur les méthodes divisives	47
7.1.1. Algorithme-type d'une méthode divisive	48
7.1.2. Choix de la classe à diviser	48
7.1.3. Division de la classe	49
7.1.4. Intérêts et désavantages des méthodes divisives	49
7.2. Extension du critère d'inertie	50
7.3. Choix de la classe à diviser	51
7.4. Bipartitionnement d'une classe	53
7.4.1. Questions binaires et objets symboliques	53
7.4.2. L'ensemble des questions binaires	56
7.4.3. Choix d'une question binaire	56
7.5. Complexité	60
7.6. Algorithmes	60
7.6.1. Mise à jour du critère d'inertie	60
7.6.2. Choix de la meilleure coupure d'une classe	62
7.6.3. Algorithme principal	63
8. Méthode des nuées dynamiques (DYCLASS)	65
8.1. Introduction	65
8.2. Principes de la méthode	66
8.3. Construction de la méthode	66
8.4. Le cas des intervalles	69
8.4.1. L'espace de représentation	69
8.4.2. Le critère de minimisation	69
8.4.3. Construction de l'algorithme	71
8.5. Algorithme MND pour les données intervalles	72

Partie IV. Programmes et résultats	73
9. Interprétation d'une partition	74
9.1. Inertie expliquée d'une partition	74
9.2. Comparaison de partitions	75
9.2.1. Le tableau de contingence	75
9.2.2. Le χ^2 d'indépendance	75
9.2.3. Maximisation "ligne-colonne" du tableau de contingence	76
10. Application : données réelles	77
10.1. Jeu de données "temp1988.sds"	77
10.1.1. La méthode divisive de Chavent	77
10.1.2. Méthode des nuées dynamiques	80
10.1.3. Comparaison des méthodes	83
11. Application : données artificielles	85
11.1. Création de données artificielles	85
11.2. Jeu de données "artificielles.sds"	93
11.2.1. La méthode divisive de Chavent	93
11.2.2. Méthode des nuées dynamiques	101
11.2.3. Comparaison des méthodes	109
Conclusion	111
Annexes	113
Annexe A	113
Annexe B : Programmes	115
Programme "comparaison.cpp"	115
Programme "données.cpp"	117
Annexe C : Données réelles	120
Fichier "temp1988.sds"	120
Fichier "chavent.txt"	124
Fichier "scluster.txt"	125

Annexe D : Données artificielles	126
Fichier "liste.txt"	126
Fichier "artificielles.sds"	138
Fichier "chavent.txt"	157
Fichier "scluster.txt"	163
Bibliographie	170

INTRODUCTION

L'analyse de données est devenue, au cours de ces dernières années, un outil indispensable en statistiques. En effet, l'expansion de l'informatique nous donne la possibilité de stocker et de gérer des bases de données toujours plus conséquentes. L'analyse de données est dès lors devenue un moyen de synthétiser ces informations en vue d'une compréhension plus aisée de celles-ci.

La classification automatique est l'une des méthodes de l'analyse de données. Elle permet de diviser une population d'individus (ou objets) décrits par un ensemble de caractéristiques (ou variables) en des sous-populations homogènes. La détermination du nombre de ces classes est, en soi-même, un problème très largement étudié et que nous n'aborderons pas. Nous nous intéresserons plutôt à deux méthodes de classification : la méthode divisive de classification hiérarchique développée par Chavent [6] et la méthode des nuées dynamiques développée par Diday [7] et adaptée aux données symboliques par Lechevallier [3]. Ces méthodes seront étudiées dans le cadre de données particulières : les données symboliques ⁽¹⁾ intervalles. Ces intervalles de valeurs ont en effet la particularité intéressante de pouvoir traduire la variation d'un individu, en ne transformant pas nécessairement sa description en une modalité unique qui

⁽¹⁾Les données symboliques font actuellement l'objet d'un projet européen d'étude du nom de SODAS.

engendrerait une perte d'information.

Le plan du travail se résume principalement en quatre grandes parties.

D'abord, nous donnerons un bref aperçu des bases de l'analyse de données en expliquant quelques notions élémentaires, en introduisant les données symboliques et en rappelant les notions d'inertie, de similarité et de dissimilarité.

Ensuite, nous développerons les grands principes de la classification et la particularité des données intervalles.

Puis, nous étudierons séparément les deux méthodes : la méthode divisive de Chavent et la méthode des nuées dynamiques (Diday) et son adaptation aux données symboliques (Lechevallier).

Enfin, nous appliquerons ces méthodes à un jeu de données réelles ainsi qu'à un jeu de données artificielles ; nous essaierons alors de distinguer les points communs et les différences des méthodes.

PARTIE I

BASES DE L'ANALYSE DE DONNÉES

CHAPITRE 1

DONNÉES CLASSIQUES

1.1. Individus et variables

On définit habituellement les relations reliant les individus et les variables de la manière suivante.

Considérons :

- ◇ un ensemble de n individus $\Omega = \{1, \dots, n\}$,
- ◇ Y_1, \dots, Y_p les p caractéristiques ou variables pour chaque individu,
- ◇ \mathcal{Y}_j l'ensemble des observations pour la variable Y_j ($j = 1, \dots, p$).

Une variable peut alors être définie comme :

- ◇ une application de Ω dans l'ensemble d'arrivée \mathcal{Y}_j .
- ◇ une structure algébrique S sur \mathcal{Y}_j pouvant se transporter sur Ω par cette application.

On peut représenter ces relations comme suit :

$$\begin{aligned} Y_j : \Omega &\rightarrow \mathcal{Y}_j \\ k &\rightsquigarrow Y_j(k) = x_{kj} \end{aligned}$$

où x_{kj} représente la valeur de la propriété j pour l'individu k .

1.2. Types de variables

Après avoir choisi les variables, il faut leur associer un type. Suivant la structure S , on distingue deux types de variables : les variables quantitatives et qualitatives.

1.2.1. Variables quantitatives

Une variable Y est dite quantitative lorsque l'ensemble des observations ou des valeurs possibles \mathcal{Y} est inclus ou égal à \mathbb{R} . Les variables quantitatives peuvent être continues ou discrètes.

On distingue les types de variables quantitatives suivants :

- ◇ quantitatif mesurable (revenu, poids, ...),
- ◇ quantitatif d'ordre (note, rang, ...),
- ◇ quantitatif de comptage (fréquence, contingence, ...),
- ◇ quantitatif binaire (succès-échec, ...).

1.2.2. Variables qualitatives

Une variable Y est dite qualitative lorsque l'ensemble d'arrivée \mathcal{Y} est fini. Ses éléments sont appelés modalités de la variable. La structure S de \mathcal{Y} n'est par conséquent ni additionnelle, ni multiplicative.

On distingue essentiellement les types de variables qualitatives suivants :

- ◇ qualitatif nominal (lieu géographique, catégorie socio-professionnelle,...) : on ne considère que la structure d'ensemble, l'application $Y : \Omega \rightarrow \mathcal{Y}$ définit ainsi une relation d'équivalence sur Ω ($\forall x, y \in \mathcal{Y} : x = y$ ou $x \neq y$) ;
- ◇ qualitatif ordinal (mauvais, insuffisant, satisfaisant, ...) : \mathcal{Y} est muni d'une structure d'ordre total ($\forall x, y \in \mathcal{Y} : x < y$ ou $x > y$).

1.3. Choix de la similarité

Le choix de la similarité (ou de la dissimilarité) est un choix important qui s'impose dans la plupart des méthodes de classification. Nous présenterons ultérieurement en détails les choix possibles : indices de similarité, de dissimilarité, distances, ultramétriques, ...

1.4. Matrice de données

Une matrice de données est une matrice rectangulaire qui se déduit de la définition de l'ensemble des individus et des variables.

Considérons l'ensemble des n individus $\Omega = \{1, \dots, n\}$ et les p variables qui y sont associées Y_1, \dots, Y_p . Les variables Y_j prennent, comme nous l'avons vu, leurs valeurs dans les ensembles d'observations \mathcal{Y}_j ($j = 1, \dots, p$).

Si l'on note X le vecteur des p variables Y_1, \dots, Y_p ; on a :

$$X = \begin{pmatrix} Y_1 \\ \vdots \\ Y_p \end{pmatrix} = (Y_1, \dots, Y_p)' \in \mathfrak{X} := \bigotimes_{j=1}^p \mathcal{Y}_j$$

où \mathfrak{X} est le produit cartésien correspondant aux domaines spécifiques $\mathcal{Y}_1, \dots, \mathcal{Y}_p$ des variables.

Le vecteur X peut être considéré comme une application de l'ensemble Ω des individus dans l'espace-produit \mathfrak{X} .

On note, comme déjà vu, $Y_j(k) = x_{kj}$:

$$\begin{aligned} Y_j : \Omega &\rightarrow \mathcal{Y}_j \\ k &\rightsquigarrow Y_j(k) = x_{kj} \end{aligned}$$

CHAPITRE 2

INTRODUCTION AUX DONNÉES SYMBOLIQUES

Nous venons de décrire la matrice des données classiques, c'est-à-dire une matrice rectangulaire $\tilde{X} = (x_{kj})_{n \times p}$ qui définit la relation entre l'ensemble $\Omega = \{1, \dots, n\}$ des n individus et une série de variables Y_1, \dots, Y_p où chaque variable Y_j contient une seule valeur de l'ensemble \mathcal{Y} (i.e. chaque cellule (k, j) de la matrice \tilde{X} ne contient qu'une seule valeur). Cependant, il existe d'autres types de variables appelées symboliques. Nous nous restreindrons ici aux variables multivaluées et plus particulièrement à un de leurs cas particuliers, les intervalles, car les autres types de variables ne feront pas l'objet des applications.

Redéfinissons également l'ensemble des "objets" E de deux façons différentes :

- ◊ Un univers $E = \Omega = \{1, \dots, n\}$ d'individus aussi appelés objets du premier ordre.
- ◊ Un système $E = \{C_1, C_2, \dots\}$ de classes $C_i \subseteq \Omega$ d'individus, c'est-à-dire des objets agrégés, également appelés objets du second ordre.

Cette terminologie se réfère directement à l'approche SODAS (Symbolic Official Data Analysis System) [2].

Une variable Y définie pour tous les éléments k d'un ensemble E est évaluée dans le domaine \mathcal{Y} si elle prend sa valeur $Y(k)$ dans le système $\mathcal{B} = \mathcal{P}(\mathcal{Y}) = \{U \neq \emptyset \mid U \subseteq \mathcal{Y}\}$

$$\begin{aligned} \forall k \in E \quad Y : E &\rightarrow \mathcal{B} \\ k &\rightsquigarrow Y(k) . \end{aligned}$$

Dans le cas des données classiques, on a donc toujours l'égalité $\forall k \in E :$

$$|Y(k)| = 1 .$$

2.1. Données multivaluées

Une variable Y est dite à valeurs multiples lorsque les valeurs $Y(k)$ sont toutes des sous-ensembles finis de \mathcal{Y} :

$$|Y(k)| < \infty .$$

Si les éléments de \mathcal{Y} sont des catégories, nous parlerons de variables à catégories multiples et, à fortiori, tous les $Y(k)$ sont finis.

Si les éléments de \mathcal{Y} sont des nombres réels ($Y(k) \in \mathbb{R}$ et $|Y(k)| < \infty$), nous parlerons de variables à valeurs quantitatives multiples.

2.2. Variables intervalles

$$\begin{aligned} Y : E &\rightarrow \mathcal{B} \\ k &\rightsquigarrow Y(k) = U \end{aligned}$$

où

- ◊ U est un intervalle de \mathbb{R} (fermé ou non),
- ◊ \mathcal{B} est l'ensemble de tous les intervalles \mathcal{I} de \mathcal{Y} .

Exemple :

- ◊ $E = \{ \text{un ensemble de villes sélectionnées dans un pays} \}$.
- ◊ $Y = \text{la température mesurée durant un mois.}$

- ◇ $\mathcal{Y} = \{[a, b] \mid a, b \in \mathbb{R}, -\infty < a < b < \infty\}$.
- ◇ Valeurs possibles : $Y(k) = [-7.3, 12.2], Y(l) = [2.1, 18.9] \dots$

Une variable quantitative de type intervalle peut être obtenue par agrégation.

Exemple :

- ◇ $E = \{C_1, \dots, C_m\}$ est l'ensemble des m classes C_i d'une école avec $\Omega = \sum_{i=1}^m C_i$ l'ensemble de tous les élèves.
- ◇ $\tilde{Y}(k)$ est la taille des élèves en mètres, une variable classique "univaluée".
- ◇ Une classe C_i est maintenant caractérisée par son étendue : $Y(C_i) = [\min, \max] = [\alpha, \beta]$, un intervalle de $\mathcal{Y} = \mathbb{R}^+$ avec
$$\begin{cases} \alpha = \min_{\omega \in C_i} \{\tilde{Y}(\omega)\} \\ \beta = \max_{\omega \in C_i} \{\tilde{Y}(\omega)\} \end{cases}.$$
- ◇ Par exemple, $Y(C_i) = [1.60, 1.80]$.

On obtient à partir de la variable \tilde{Y} une variable quantitative intervalle.

En résumé, on a créé une variable agrégée Y non seulement pour la classe $C_i \in E$ mais pour tous les sous-ensembles $C \subseteq \mathcal{P}(\Omega)$ dont toutes les valeurs prises sont des sous-ensembles $Y(C) \subseteq \mathcal{Y}$.

2.3. Matrice des données symboliques

Par la suite, on appellera $X(u) = (Y_1(u), \dots, Y_p(u))'$ le vecteur des variables symboliques obtenues pour $u \in E$. Dès lors, chaque individu $u \in E$ peut être décrit par un vecteur de données symboliques :

$$x_u = X(u) = \begin{pmatrix} \xi_{u1} \\ \vdots \\ \xi_{up} \end{pmatrix} = (\xi_{u1}, \dots, \xi_{up})' \in \mathfrak{X} = \bigotimes_{j=1}^p \mathcal{B}_j.$$

CHAPITRE 3

NOTION D'INERTIE

Nous avons vu qu'un tableau de données à n individus et p variables quantitatives définissait un ensemble de n points dans \mathbb{R}^p . On peut identifier l'ensemble Ω des individus et l'ensemble des n points de \mathbb{R}^p :

$$\Omega = (x_1, \dots, x_n) \text{ avec } x_i = \begin{pmatrix} x_i^1 \\ \vdots \\ x_i^p \end{pmatrix} \in \mathbb{R}^p .$$

D'autre part, chacun de ces n points est muni d'un poids p_i strictement positif et de somme habituellement égale à 1 ($\sum_{i=1}^n p_i = 1$). Nous pouvons donc construire un nuage $N(\Omega)$ de n points pondérés dans \mathbb{R} :

$$N(\Omega) = \{(x_i, p_i) , i = 1, \dots, n\} .$$

3.1. Inertie du nuage par rapport à un point

L'espace \mathbb{R}^p étant muni d'une métrique euclidienne d_M^2 (définie par la relation $d_M^2(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j)$ où M est symétrique et définie positive), on appelle inertie de $N(\Omega)$ par rapport à un point a de \mathbb{R}^p la quantité :

$$I_a = \sum_{i=1}^n p_i d_M^2(x_i, a) .$$

3.2. Théorème de Huygens

Si $g = \sum_{i=1}^n p_i x_i$ est le centre de gravité du nuage $N(\Omega)$ on a :

$$\forall a \in \mathbb{R}^p, I_a = I_g + \sum_{i=1}^n p_i d_M^2(a, g) .$$

Démonstration :

$$\forall a \in \mathbb{R}^p, I_a = \sum_{i=1}^n p_i (x_i - a)^T M (x_i - a) .$$

$$\begin{aligned} I_a &= \sum_{i=1}^n p_i (x_i - g + g - a)^T M (x_i - g + g - a) \\ &= \sum_{i=1}^n p_i (x_i - g)^T M (x_i - g) + 2 \sum_{i=1}^n p_i (g - a)^T M (x_i - g) \\ &\quad + \sum_{i=1}^n p_i (g - a)^T M (g - a) \text{ (car } M \text{ est symétrique)} \\ &= I_g + \sum_{i=1}^n p_i d_M^2(a, g) + 2 (g - a)^T M \sum_{i=1}^n p_i (x_i - g) \\ &= I_g + \sum_{i=1}^n p_i d_M^2(a, g) \end{aligned}$$

car $\sum_{i=1}^n p_i (x_i - g) = \sum_{i=1}^n p_i x_i + \sum_{i=1}^n p_i g = g - g = 0$, par définition du centre de gravité et de la somme des poids.

□

Le centre de gravité g est donc le point par rapport auquel l'inertie du nuage est minimale. I_g est l'inertie totale du nuage $N(\Omega)$ et sera souvent notée T (Total Inertia).

3.3. Inertie associée à une partition

Soit $\mathcal{C} = (C_1, \dots, C_m)$ une partition en m classes de Ω . Nous notons :

$$\mu_k = \sum_{x_i \in C_k} p_i \text{ le poids de la classe } C_k, \quad k = 1 \dots m.$$

Nous gardons la propriété de la somme des poids :

$$\sum_{k=1}^m \mu_k = 1$$

et

$$g_k = \frac{1}{\mu_k} \sum_{x_i \in C_k} p_i x_i$$

est le centre de gravité de la classe C_k , $i = 1 \dots m$.

3.3.1. Les trois types d'inerties

A cette partition \mathcal{C} , nous pouvons maintenant associer trois types d'inerties :

◊ L'inertie totale T (Total Inertia), indépendante de la partition :

$$T = \sum_{i=1}^n p_i d_M^2(x_i, g) = \sum_{i=1}^n p_i (x_i - g)^T M (x_i - g) .$$

◊ L'inertie inter-classe B (Between-Class Inertia)

$$B = \sum_{k=1}^m \mu_k d_M^2(g_k, g) = \sum_{k=1}^m \mu_k (g_k - g)^T M (g_k - g) .$$

◊ L'inertie intra-classe W (Within-Class Inertia)

$$W = \sum_{k=1}^m W_k ,$$

où W_k est l'inertie de la classe C_k par rapport à son propre centre de gravité g_k :

$$W_k = \sum_{x_i \in C_k} p_i d_M^2(x_i, g_k)$$

d'où

$$W = \sum_{k=1}^m \sum_{x_i \in C_k} p_i (x_i - g_k)^T M (x_i - g_k) .$$

3.3.2. Relation entre les inerties partielles

Il existe une relation fondamentale reliant ces trois types d'inerties :

$$T = B + W .$$

Cette relation se déduit du théorème d'Huygens comme suit :

Démonstration :

Si nous considérons l'ensemble des points d'une des classes C_k , d'après le théorème d'Huygens, nous pouvons définir l'inertie du nuage de points associée à cette classe par rapport au point g comme suit :

$$I_g = I_{g_k} + \sum_{x_i \in C_k} p_i d_M^2(g, g_k) .$$

On peut réécrire cette équation en la décomposant comme suit :

$$\sum_{x_i \in C_k} p_i d_M^2(x_i, g) = \sum_{x_i \in C_k} p_i d_M^2(x_i, g_k) + \mu_k d_M^2(g_k, g) .$$

En sommant cette égalité pour k variant de 1 à m , nous obtenons la somme $T = W + B$ désirée. Il résulte donc de cette formule que plus l'inertie intra-classe est faible plus l'inertie inter-classe est grande et inversement.

□

3.3.3. Décomposition des inerties sur les classes et les variables

Il est également utile de décomposer les inerties associées suivant les classes et les variables, cela nous permettra de donner une interprétation d'une partition. On supposera que la métrique M est diagonale (en effet, on choisit en général $M = I$).

◇ T peut s'écrire :

$$T = \sum_{k=1}^m \sum_{j=1}^p T_k^j$$

où

$$T_k^j = \sum_{x_i \in C_k} p_i M_{jj} (x_i^j - g^j)^2 ,$$

M_{jj} étant le terme général de la diagonale de la matrice M .

T_k^j est l'écart moyen, pour la variable j , des points de la classe C_k au centre de gravité global.

◇ B peut s'écrire :

$$B = \sum_{k=1}^m \sum_{j=1}^p B_k^j$$

où

$$B_k^j = \sum_{x_i \in C_k} p_i M_{jj} (g_k^j - g^j)^2 = \mu_k M_{jj} (g_k^j - g^j)^2 .$$

B_k^j est l'écart, pour la variable j , du centre de gravité de la classe C_k au centre de gravité global.

◇ W peut s'écrire :

$$W = \sum_{k=1}^m \sum_{j=1}^p W_k^j$$

où

$$W_k^j = \sum_{x_i \in C_k} p_i M_{jj} (x_i^j - g_k^j)^2 .$$

W_k^j est l'inertie de la classe C_k pour la variable j .

Les différentes inerties se décomposent de manière additive sur les variables et sur les classes, ce que nous pouvons écrire :

◇ $\forall j = 1, \dots, p :$

$$T^j = \sum_{k=1}^m T_k^j , \quad B^j = \sum_{k=1}^m B_k^j , \quad W^j = \sum_{k=1}^m W_k^j$$

et ensuite

$$T = \sum_{j=1}^p T^j, \quad B = \sum_{j=1}^p B^j, \quad W = \sum_{j=1}^p W^j.$$

T^j , B^j et W^j expriment la part de la variable j pour les inerties concernées.

◇ $\forall k = 1, \dots, m$:

$$T_k = \sum_{j=1}^p T_k^j, \quad B_k = \sum_{j=1}^p B_k^j, \quad W_k = \sum_{j=1}^p W_k^j$$

et ensuite

$$T = \sum_{k=1}^m T_k, \quad B = \sum_{k=1}^m B_k, \quad W = \sum_{k=1}^m W_k.$$

T_k , B_k et W_k expriment la part de la classe C_k pour les inerties concernées.

Nous voyons qu'en fait la relation $T = B + W$ se conserve par décomposition selon j et m . On a :

$$\begin{array}{ll} \forall j = 1 \dots p \text{ et } \forall k = 1 \dots m & T_k^j = B_k^j + W_k^j, \\ \forall j = 1 \dots p & T^j = B^j + W^j, \\ \forall k = 1 \dots m & T_k = B_k + W_k. \end{array}$$

Ces relations permettront une interprétation "inertielle" d'une partition.

CHAPITRE 4

SIMILARITÉS ET DISSIMILARITÉS

4.1. Indices de similarité et de dissimilarité

La classification est basée sur la similarité qui peut exister entre des individus afin de créer des sous-ensembles ou classes d'objets apparentés.

L'objectif est donc de rechercher des classes $C_1, C_2, \dots \subseteq \Omega$ homogènes d'objets, telles que la similarité soit grande entre les paires d'objets appartenant à la même classe et faible entre les paires appartenant à des classes différentes.

4.1.1. Mesure de ressemblance

Soit E un ensemble d'objets pour lesquels nous voulons définir une mesure de ressemblance. E peut être de différents types :

- ◇ $E = \Omega = \{1, \dots, n\}$ un ensemble d'individus ou d'objets,
- ◇ $E \subseteq \Omega$ un sous-ensemble,
- ◇ $E = \{C_1, \dots, C_m\}$ une collection de classes $C_1, \dots, C_m \in \Omega$,
- ◇ E l'ensemble-solution \mathcal{Y}_j pour la variable Y_j ,
- ◇ $E = \{x_1, \dots, x_n\}$ un ensemble de n vecteurs de données caractérisant les n individus.

Une mesure de ressemblance sur E est une fonction $r(.,.)$ à valeurs réelles positives qui est définie pour toutes paires (a, b) d'éléments de E :

$$\begin{aligned} r : E \times E &\rightarrow \mathbb{R}^+ \\ (k, l) &\rightsquigarrow r(k, l) . \end{aligned}$$

Une mesure de ressemblance quelconque devra également vérifier la condition de symétrie :

$$\forall a, b \in E : r(a, b) = r(b, a) .$$

4.1.2. Indice de similarité

En plus de la condition de symétrie, l'indice de similarité est une mesure de ressemblance $s(.,.)$ ayant comme propriété :

$$\forall (a, b) \in E \times E, a \neq b : s(a, a) = s(b, b) > s(a, b)$$

Nous remarquons que la similarité d'un point à lui-même est une constante supérieure à tout autre similarité : nous la notons s_{\max} et nous l'égalons souvent à 1 (normalisation).

En conclusion, un indice de similarité normalisé vérifie la condition suivante :

$$0 \leq s(b, a) = s(a, b) \leq s(a, a) = 1 \quad \forall a, b \in E .$$

4.1.3. Indice de dissimilarité

L'indice de dissimilarité est, quant à lui, une mesure de ressemblance $d(.,.)$ ayant comme propriété supplémentaire :

$$\forall a \in E : d(a, a) = 0 .$$

En conclusion, un indice de dissimilarité vérifie la condition suivante :

$$0 \leq d(a, a) \leq d(a, b) = d(b, a) < \infty \quad \forall a, b \in E .$$

4.1.4. Transformation d'un indice de similarité en indice de dissimilarité et vice-versa

Nous pouvons transformer un indice de similarité en un indice de dissimilarité et inversement en posant :

- ◇ $d = \phi(s)$ où $\phi(\cdot)$ est une fonction strictement décroissante telle que $\phi(0) = \infty$ et $\phi(1) = 0$,
- ◇ $s = \psi(d)$ où $\psi(\cdot)$ est une fonction strictement décroissante telle que $\psi(0) = 1$ et $\psi(\infty) = 0$.

Par exemple, les transformations ψ les plus couramment utilisées sont :

- ◇ $s = \max(d) - d$,
- ◇ $s = \sqrt{\max(d) - d}$,
- ◇ $s = \max(d^2) - d^2$,

où $\max(d)$ est la valeur maximale (observée ou observable) de l'indice d .

4.1.5. Semi-ordre généré par une mesure de ressemblance

Une mesure de ressemblance r sur E (en particulier un indice de similarité ou de dissimilarité) induit un semi-ordre \preceq_r sur l'ensemble des paires ordonnées de $E \times E$:

$$\forall (a, b), (c, d) \in E \times E :$$

$$(a, b) \preceq_r (c, d) \text{ si } \begin{cases} d(a, b) \leq d(c, d) & \text{si } r \equiv d \\ s(a, b) \geq s(c, d) & \text{si } r \equiv s . \end{cases}$$

Nous pouvons interpréter ce résultat de la manière suivante : les éléments du couple (a, b) se ressemblent plus que les éléments du couple (c, d) .

Par la définition du semi-ordre \preceq_r , nous observons également les propriétés suivantes :

◇ réflexivité :

$$\forall a, b \in E : (a, b) \preceq_r (a, b) .$$

◇ transitivité :

$$\begin{aligned} &\forall (a, b), (c, d), (e, f) \in E \times E, \\ &(a, b) \preceq_r (c, d) \wedge (c, d) \preceq_r (e, f) \Rightarrow (a, b) \preceq_r (e, f) . \end{aligned}$$

Deux mesures de ressemblance r et r' seront équivalentes si et seulement les semi-ordres correspondants \preceq_r et $\preceq_{r'}$ sur $E \times E$ sont identiques.

4.2. Indices de dissimilarité

4.2.1. Matrice de dissimilarité

Nous avons vu qu'il y a toujours moyen de transformer un indice de similarité en indice de dissimilarité, nous allons dorénavant nous restreindre aux dissimilarités.

Soit, donc, d un indice de dissimilarité sur l'ensemble E tel que :

$$0 \leq d(a, a) \leq d(a, b) = d(b, a) < \infty \quad \forall a, b \in E .$$

Si $E = \Omega$ est un ensemble de n individus, nous pouvons définir la matrice de dissimilarité $D = (d_{kl})$ ($k, l = 1, \dots, n$) où d_{kl} est la dissimilarité entre deux individus $k, l \in \Omega$.

Cette matrice de dissimilarité D possède les particularités suivantes :

- ◇ D est symétrique puisque $d_{kl} = d(k, l) = d(l, k) = d_{lk}$,
- ◇ Les éléments de D sont tous positifs vu que $d_{kl} = d(k, l) \geq 0$,
- ◇ Les éléments diagonaux de D sont nuls vu que $d_{kk} = d(k, k) = 0$.

4.2.2. Propriétés de différents indices de dissimilarité

Si nous ajoutons d'autres contraintes sur notre indice de dissimilarité, nous pouvons obtenir :

- ◇ Une dissimilarité d vérifiant l'inégalité triangulaire :

$$\forall a, b, c \in E : d(a, b) \leq d(a, c) + d(c, b)$$

est appelée semi-distance sur E ou encore pseudo-métrique.

- ◇ Une dissimilarité d vérifiant l'inégalité triangulaire ainsi que la propriété d'unicité :

$$\forall a, b \in E : d(a, b) = 0 \Rightarrow a = b$$

est appelée distance ou métrique. Nous pouvons remarquer alors que la matrice de dissimilarité possède une propriété supplémentaire : les éléments non-diagonaux de D sont strictement positifs.

- ◇ Une dissimilarité d vérifiant l'inégalité ultramétrique :

$$\forall a, b, c \in E : d(a, b) \leq \max\{d(a, c), d(c, b)\}$$

est appelée ultramétrique.

- ◇ Une dissimilarité d vérifiant l'inégalité de Buneman :

$$\forall a, b, c, d \in E :$$

$$d(a, b) + d(c, d) \leq \max\{d(a, c) + d(b, d), d(a, d) + d(b, c)\}$$

est utilisée comme distance pour un arbre hiérarchique.

- ◇ Une matrice de dissimilarité $D = (d_{kl})_{k,l=1,\dots,n}$ sur $E = \{1, \dots, n\}$ est dite de Robinson si les dissimilarités d_{kl} croissent lorsque k ou l s'éloigne de la diagonale ($k = l$). Nous pouvons formaliser cette proposition comme suit :

$$\forall k \in \{1, \dots, n\}$$

$$d_{k,k} \leq d_{k,k+1} \leq d_{k,k+2} \leq \dots \leq d_{k,n} \text{ et } d_{k,k} \leq d_{k,k-1} \leq d_{k,k-2} \leq \dots \leq d_{k,1}$$

$$d_{k,k} \leq d_{k+1,k} \leq d_{k+2,k} \leq \dots \leq d_{n,k} \text{ et } d_{k,k} \leq d_{k-1,k} \leq d_{k-2,k} \leq \dots \leq d_{1,k}$$

4.3. Mesures de distance

Pour déterminer le degré de dissimilarité entre les paires d'individus (k, l) d'un ensemble $\Omega = \{1, \dots, n\}$ de n individus, nous devons construire la matrice

de dissimilarité $D = (d_{kl})$ telle que les valeurs d_{kl} reflètent bien les dissimilarités qu'il existe entre les individus de Ω .

La plupart du temps, la dissimilarité d_{kl} doit être calculée à partir de la matrice des données $\tilde{X} = (x_{kl})_{n \times p}$, c'est-à-dire à partir des vecteurs de données x_k, x_l se trouvant dans l'ensemble $\mathfrak{X} = \prod_{j=1}^p \mathcal{Y}_j$. Cependant, rappelons que les dissimilarités d_{kl} peuvent s'écrire $d(k, l)$ où $d(., .)$ est une mesure définie sur $\mathfrak{X} = \prod_{j=1}^p \mathcal{Y}_j$.

Cet indice de dissimilarité d peut être facilement obtenu à partir des mesures de ressemblance δ_j définies sur \mathcal{Y}_j pour chaque variable Y_j ($j = 1, \dots, p$). Cet indice total peut être calculé comme suit :

$$d_{kl} = d(x_k, x_l) = \sum_{j=1}^p \delta_j(x_{kj}, x_{lj})$$

et s'il faut introduire des poids $\omega_j \geq 0$, $j = 1, \dots, p$:

$$d_{kl} = d(x_k, x_l) = \sum_{j=1}^p \omega_j \delta_j(x_{kj}, x_{lj})$$

4.3.1. Indices de dissimilarité pour les individus

Comme illustration des définitions, nous pouvons rappeler quelques indices de dissimilarité pour les individus qui sont typiquement utilisés en analyse de données.

◇ La distance euclidienne générale :

La distance euclidienne générale doit vérifier la condition :

$$d_{kl} = d_M(x_k, x_l) = \sqrt{(x_k - x_l)' M (x_k - x_l)}$$

où M est une matrice symétrique définie positive. On nomme ces distances quadratiques ou métriques.

◇ La distance euclidienne simple sur \mathbb{R}^p :

Pour une matrice de données $\tilde{X} = (x_{kl})$ avec p variables quantitatives, si nous connaissons $x_k, x_l \in \mathbb{R}^p$, nous pouvons utiliser la distance métrique euclidienne où $M = I$ définie sur \mathbb{R}^p comme suit :

$$d_{kl} = d_2(x_k, x_l) = \|x_k - x_l\|_2 = \sqrt{\sum_{j=1}^p (x_{kj} - x_{lj})^2} .$$

◇ La distance de Minkowski :

Soit un nombre réel $q \geq 1$, la distance de Minkowski ou distance L_q sur \mathbb{R}^p est définie comme suit :

$$d_{kl} = d_q(x_k, x_l) = \|x_k - x_l\|_q = \left[\sum_{j=1}^p (x_{kj} - x_{lj})^q \right]^{\frac{1}{q}} .$$

On remarque que :

◇ pour $q = 1$, on retrouve la distance des valeurs absolues appelée également distance "city block" ou de Manhattan :

$$d_{kl} = d_1(x_k, x_l) = \|x_k - x_l\|_1 = \sum_{j=1}^p |x_{kj} - x_{lj}| ,$$

◇ pour $q = 2$, on retrouve la distance euclidienne simple,

◇ pour $q \rightarrow +\infty$, on retrouve la distance de Tchebycheff :

$$d_{kl} = d_\infty(x_k, x_l) = \|x_k - x_l\|_\infty = \max_{j=1}^p |x_{kj} - x_{lj}| .$$

◇ La distance de Mahalanobis :

Dans le cas où nous avons p variables quantitatives, un cas particulier de la distance euclidienne générale est la distance de Mahalanobis où $M = \Sigma^{-1}$ est l'inverse de la matrice de covariance pour les variables Y_1, \dots, Y_p :

$$d_{kl} = d_{\Sigma^{-1}}(x_k, x_l) = \|x_k - x_l\|_{\Sigma^{-1}} = \sqrt{(x_k - x_l)' \Sigma^{-1} (x_k - x_l)} .$$

Cet indice de dissimilarité pondéré peut être utilisé si nous désirons prendre en compte une dépendance statistique parmi les p variables quantitatives Y_1, \dots, Y_p observées.

La matrice Σ est calculée à partir des données en vecteurs $x_1, \dots, x_n \in \mathbb{R}^p$ des n individus comme suit :

$$\Sigma = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})'$$

où $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ est le centre de gravité de l'ensemble des points.

Dans le cadre de l'analyse d'une classification, nous pouvons également définir la matrice de covariance entre les classes par :

$$\Sigma = \frac{1}{n-m} \sum_{k=1}^m \sum_{x_i \in C_k} (x_i - \bar{x}_{C_k})(x_i - \bar{x}_{C_k})'$$

où $\mathcal{C} = \{C_1, \dots, C_m\}$ est une partition de l'ensemble $E = \{1, \dots, n\}$ des individus en un certain nombre m de classes C_1, \dots, C_m avec \bar{x}_{C_k} leurs centroïdes respectifs.

◇ La distance du χ^2 :

La distance du χ^2 est une distance importante qui est également utilisée en analyse de données, elle se définit comme suit :

$$\begin{aligned} d_{kl} &= d_{\chi^2}(x_k, x_l) = \|x_k, x_l\|_{\chi^2} \\ &= \left[\sum_{j=1}^p \frac{1}{\sum_{i=1}^n x_{ij}} \left(\frac{x_{kj}}{\sum_{j=1}^p x_{kj}} - \frac{x_{lj}}{\sum_{j=1}^p x_{lj}} \right)^2 \right]^{\frac{1}{2}}. \end{aligned}$$

PARTIE II

PRINCIPES DE CLASSIFICATION

CHAPITRE 5

INTRODUCTION À LA CLASSIFICATION

5.1. Problème de classification

La nature offre un grand nombre de populations qu'il est souhaitable de répartir en catégories. Les disciplines scientifiques sollicitent des classifications :

- ◇ En médecine, regroupements de malades selon certains paramètres observés sur chaque individu.
- ◇ En botanique, mise en évidence de sous-espèces d'une même variété par caractéristiques semblables.
- ◇ En reconnaissance des formes, mise en évidence des types d'empreintes digitales, d'écritures.
- ◇ ...

Les progrès informatiques ont permis ces dernières années de créer et d'améliorer des algorithmes de classification de plus en plus puissants et pouvant s'appliquer à des ensembles de données toujours plus vastes.

Les méthodes de classification peuvent être classées selon divers critères comme :

- ◇ Le type de données,
- ◇ Le type de critère de la méthode de classification,
- ◇ Le type de la structure de classification,
- ◇ Le type d'algorithme,
- ◇ ...

5.2. Principales structures de classification

Nous serons amenés, par le principe-même de la classification, à rechercher une structure inter-classe. Définissons quelques unes de ces structures puis montrons qu'elles ont des propriétés communes que nous pouvons formaliser.

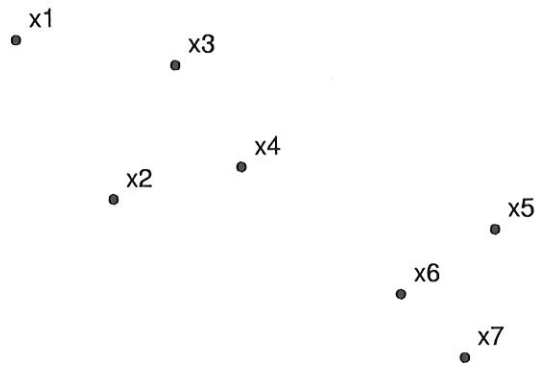


FIGURE 1. Les données

5.2.1. Partitions

Une partition d'un ensemble Ω est un ensemble de parties non-vides $\mathcal{C} = (C_1, \dots, C_m)$ d'intersections vides deux à deux et dont la réunion forme Ω :

- ◇ $\forall i \in \{1, \dots, m\} : C_i \neq \emptyset$,
- ◇ $\forall i, j \in \{1, \dots, m\}, i \neq j : C_i \cap C_j = \emptyset$,
- ◇ $\bigcup_{i=1}^m C_i = \Omega$.

Exemple :

Étant donné les points du plan de la Figure 1, on peut construire une partition en trois classes $\mathcal{C} = (C_1, C_2, C_3)$ qui est représentée par la Figure 2, avec $C_1 = \{x_1\}$, $C_2 = \{x_2, x_3, x_4\}$ et $C_3 = \{x_5, x_6, x_7\}$.

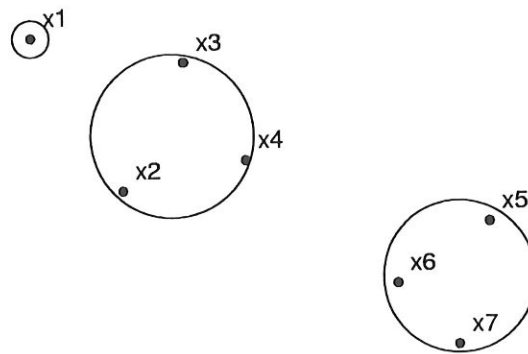


FIGURE 2. Une partition

5.2.2. Recouvrements

Un recouvrement de Ω est un ensemble $\mathcal{C} = (C_1, \dots, C_m)$ de parties non vides de Ω dont la réunion forme Ω . Une partition est donc un cas particulier de recouvrement :

- ◇ $\forall i \in \{1, \dots, m\} : C_i \neq \emptyset$,
- ◇ $\bigcup_{i=1}^m C_i = \Omega$.

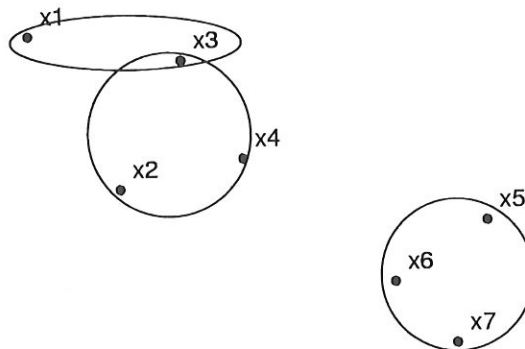


FIGURE 3. Un recouvrement

Exemple :

Étant donné les points du plan de la Figure 1, on peut construire un recouvrement en trois classes $\mathcal{C} = (C_1, C_2, C_3)$ qui est représenté par la Figure 3, avec $C_1 = \{x_1, x_3, x_4\}$, $C_2 = \{x_2, x_3, x_4\}$ et $C_3 = \{x_5, x_6, x_7\}$.

5.2.3. Hiérarchies

Nous représentons Ω par un ensemble de partitions emboîtées. En général, une hiérarchie est visualisée par un arbre appelé dendrogramme.

La définition d'une hiérarchie est la suivante : soient un ensemble Ω fini et \mathcal{H} un ensemble de parties (une collection finie de sous-ensembles non vides de Ω), \mathcal{H} est une hiérarchie sur Ω si :

- ◇ $\Omega \in \mathcal{H}$, c'est-à-dire que le palier le plus haut contient tous les individus,
- ◇ $\forall w \in \Omega : \{w\} \in \mathcal{H}$, ce sont en fait les feuilles de l'arbre,
- ◇ $\forall h, h' \in \mathcal{H} : h \cap h' \neq \emptyset \Rightarrow h \subset h' \text{ ou } h' \subset h$.

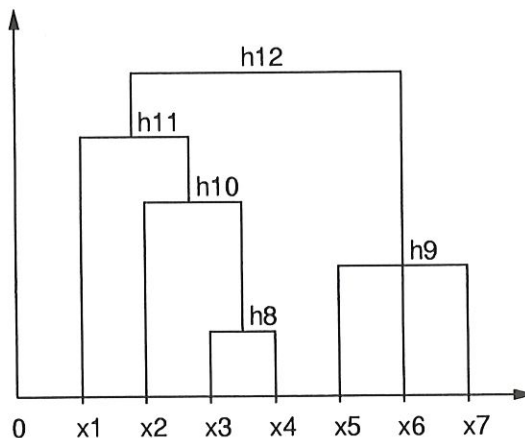


FIGURE 4. Une hiérarchie

Exemple :

Soit Ω l'ensemble des points de la Figure 1 et \mathcal{H} la hiérarchie associée (Figure 4).

On a $\mathcal{H} = \bigcup_{i=1}^{12} h_i$ avec $h_i = \{x_i\}$, $i = 1, \dots, 7$; $h_8 = \{x_3, x_4\}$, $h_9 = \{x_5, x_6, x_7\}$, $h_{10} = \{x_2\} \cap h_8$, $h_{11} = \{x_1\} \cap h_{10}$ et $h_{12} = h_9 \cup h_{11}$. Il est facilement vérifiable que \mathcal{H} satisfait aux trois axiomes de la définition.

5.2.4. Pyramides

Contrairement aux hiérarchies, cette représentation autorise la possibilité pour un singleton ou un palier d'appartenir à deux classes. Une pyramide est donc une suite de recouvrements emboîtés alors qu'une hiérarchie est une suite de partitions emboîtées.

La définition d'une pyramide est la suivante : soient Ω un ensemble fini et \mathcal{P} un ensemble de parties non vides sur Ω , \mathcal{P} est une pyramide si :

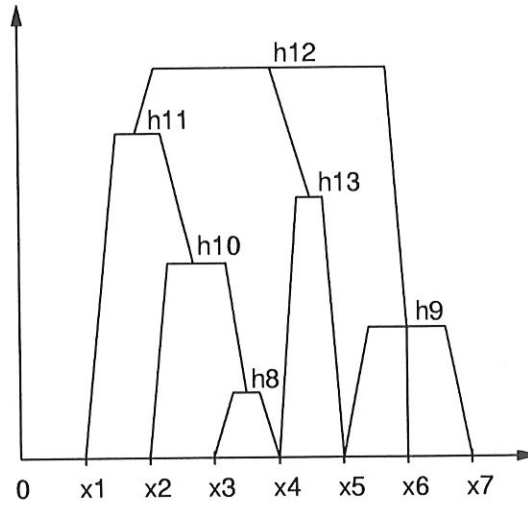


FIGURE 5. Une pyramide

- ◇ $\Omega \in \mathcal{P}$, i.e. le palier le plus haut contient tous les individus,
- ◇ $\forall w \in \Omega : \{w\} \in \mathcal{P}$, i.e. les feuilles de l'arbre,
- ◇ $\forall h, h' \in \mathcal{P} : h \cap h' = \emptyset$ ou $h \cap h' \in \mathcal{P}$,

◇ Il existe un ordre θ tel que tout élément de \mathcal{P} soit un intervalle de θ .

Exemple :

Soit la pyramide \mathcal{P} de la Figure 5. Les paliers h_i de \mathcal{P} sont les mêmes que ceux de la hiérarchie \mathcal{H} de l'exemple précédent pour $i = 1, \dots, 7$. Nous avons comme seule différence que $\mathcal{P} = \mathcal{H} \cup h_{13}$ avec $h_{13} = \{x_4, x_6\}$. En considérant que l'ordre θ est celui des singletons à la base de la pyramide \mathcal{P} , on vérifie facilement que les quatre axiomes de la définition sont satisfaits.

5.2.5. Formalisation de la notion d'espace de classification

Nous pouvons montrer que les ensembles de classes d'individus que nous venons de présenter (partitions, recouvrements, hiérarchies, pyramides) satisfont à des propriétés communes.

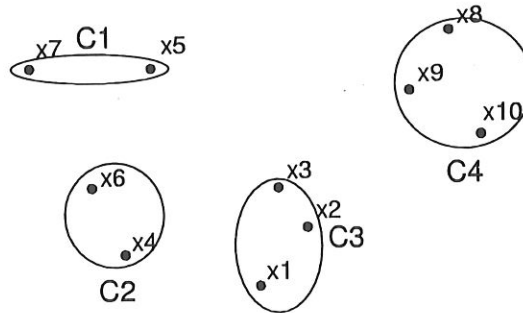


FIGURE 6. Une classification naturelle

Soit Ω l'ensemble formé des 10 points de la Figure 6 et les classes qui leur sont associées par les contours indiqués.

A partir de ces 10 points, nous pouvons définir plusieurs classifications "naturelles". Si nous considérons que tous les points forment une seule classe, nous obtenons la classification définie par l'ensemble Ω lui-même. L'ensemble des classes (C_1, C_2, C_3, C_4) définit une deuxième classification. Une troisième classification consiste à considérer que les 10 points notés x_1, \dots, x_{10} forment

10 classes réduites chacune à un élément.

Considérons maintenant l'ensemble S formé de ces trois classifications. On a donc :

$$S = \{\{\Omega\}, \{C_1, C_2, C_3, C_4\}, \{\{x_1\}, \dots, \{x_{10}\}\}\}$$

L'ensemble S satisfait aux quatre propriétés suivantes :

- ◇ S est formé de trois éléments qui forment chacun une partie de l'ensemble des parties de Ω ,
- ◇ Chacun des éléments de S est formé de classes dont la réunion recouvre Ω ,
- ◇ Un des éléments de S contient Ω ,
- ◇ Un des éléments de S contient tous les singletons.

Nous pouvons formaliser ces résultats et ainsi donner une définition rigoureuse d'un espace ou d'une structure de classification :

L'ensemble S constitue un espace de classification sur Ω si S satisfait aux quatre propriétés suivantes :

- ◇ $S \in \mathcal{P}(\mathcal{P}(\Omega))$ où $\mathcal{P}(\Omega)$ est l'ensemble des parties de Ω .
- ◇ $\forall s \in S, s = \{C_1, \dots, C_m\} : \bigcup_{i=1}^m C_i = \Omega$.
- ◇ $\exists s \in S$ tel que $\Omega \in s$.
- ◇ $\exists s \in S$ tel que $\forall w \in \Omega, \{w\} \in s$.

Proposition :

Les partitions, les recouvrements, les hiérarchies et les pyramides sont des espaces de classification.

Démonstration :

Appelons \mathcal{P} l'ensemble des partitions, on a :

$$\diamond \mathcal{P} \subset \mathcal{P}(\mathcal{P}(\Omega)).$$

$$\diamond \forall P \in \mathcal{P}, P = \{P_1, \dots, P_m\} \Rightarrow \bigcup_{i=1}^m P_i = \Omega.$$

$$\diamond \exists P \in \mathcal{P} \text{ tel que } \Omega \in P, \text{ i.e. il suffit de prendre la partition réduite à } \Omega.$$

$$\diamond \exists P \in \mathcal{P} \text{ tel que } \forall w \in \Omega, \{w\} \in P, \text{ i.e. il suffit de considérer la partition dont chaque partie est un singleton.}$$

La démonstration est analogue pour les recouvrements. Quant aux autres types de structure, rappelons qu'une hiérarchie est une suite de partitions emboîtées et qu'une pyramide est une suite de recouvrements emboîtés.

□

CHAPITRE 6

CLASSIFICATION ET DONNÉES INTERVALLES

Notre but étant d'étudier et de comparer deux méthodes de classification pour un tableau de données symboliques intervalles de taille $n \times p : \tilde{X}$.

Nous nous limiterons à la présentation de ces méthodes dans le cas de variables symboliques Y_j avec un espace d'observation \mathcal{Y}_j tel que le tableau des données contienne des variables intervalles.

6.1. Opérateur d'union jointe pour les variables intervalles

Nous avons déjà évoqué ce type de variable au chapitre 2 et notamment la manière de représenter une matrice de données symboliques (section 2.3).

Soit \mathcal{I}_A et \mathcal{I}_B deux intervalles de \mathbb{R} (i.e. $\mathcal{I}_A, \mathcal{I}_B \in \mathcal{I}$), nous voudrions appliquer les opérateurs classiques d'union et d'intersection ensemblistes : $\mathcal{I}_A \cup \mathcal{I}_B$ et $\mathcal{I}_A \cap \mathcal{I}_B$. Le problème est que l'intersection ensembliste de deux intervalles est un intervalle mais que l'union ne l'est pas nécessairement. C'est pourquoi Ichino définit l'opérateur d'union jointe :

Soient \mathcal{I}_A et \mathcal{I}_B deux éléments de \mathcal{I} . L'union jointe $\mathcal{I}_A \oplus \mathcal{I}_B$ est définie par :

$$\mathcal{I}_A \oplus \mathcal{I}_B = [\min(a_i, b_i), \max(a_s, b_s)]$$

où \mathcal{I}_A et \mathcal{I}_B sont deux intervalles $[a_i, a_s]$ et $[b_i, b_s]$.

6.2. Mesure de Hausdorff et fonction de comparaison de Ichino

6.2.1. Mesure de Hausdorff

◇ Soit δ_j une distance définie sur \mathcal{B}_j :

$$\begin{aligned} \delta_j : \mathcal{B}_j \times \mathcal{B}_j &\rightarrow \mathbb{R}^+ \\ (\mathcal{I}_A^j, \mathcal{I}_B^j) &\rightsquigarrow \delta(\mathcal{I}_A^j, \mathcal{I}_B^j) \end{aligned}$$

où $\mathcal{I}_A^j = [a_i^j, a_s^j]$ et $\mathcal{I}_B^j = [b_i^j, b_s^j]$ sont deux intervalles. Dans ce cas, nous utilisons la distance de Hausdorff :

$$\delta_j(\mathcal{I}_A^j, \mathcal{I}_B^j) = \max\{|b_i^j - a_i^j|, |b_s^j - a_s^j|\} .$$

Notons que $\delta(\mathcal{I}_A^j, \mathcal{I}_B^j)$ est alors la distance L_∞ entre les points (a_i^j, a_s^j) et (b_i^j, b_s^j) de \mathbb{R}^2 . Dans le cas de variables univaluées, la formule est réduite à la valeur absolue entre deux valeurs singletons.

◇ Les distances spécifiques δ_j pour chaque variable peuvent être combinées en une distance d définie sur Ω au moyen de la formule de composition suivante :

$$\begin{aligned} d : \Omega \times \Omega &\rightarrow \mathbb{R}^+ \\ (\mathcal{I}_A, \mathcal{I}_B) &\rightsquigarrow d(A, B) = \left(\sum_{j=1}^p \delta_j(\mathcal{I}_A^j, \mathcal{I}_B^j)^\alpha \right)^{\frac{1}{\alpha}} \end{aligned}$$

que nous pouvons réécrire autrement :

$$\begin{aligned} d : \Omega \times \Omega &\rightarrow \mathbb{R}^+ \\ (\mathcal{I}_A, \mathcal{I}_B) &\rightarrow d(A, B) = \left(\sum_{j=1}^p \max\{|b_i^j - a_i^j|, |b_s^j - a_s^j|\}^\alpha \right)^{\frac{1}{\alpha}} . \end{aligned}$$

◇ Nous pouvons également utiliser cette formule de composition afin de calculer une distance normalisée. En effet, lorsque toutes les variables sont mesurées avec une échelle métrique identique mais avec des unités

de mesures différentes, l'utilisation telle quelle des variables dans le calcul de la distance entre individus donnera plus de poids aux variables de plus forte dispersion, annihilant complètement l'effet des autres variables. Cet effet sera supprimé en normalisant les données. Dans le cas d'un tableau de données symboliques, nous pouvons également tenir compte de cet effet de pondération dans la définition de ressemblance. A partir de la formule de composition vue précédemment, nous définissons une mesure de ressemblance :

$$d: \Omega \times \Omega \rightarrow \mathbb{R}^+$$

$$(\mathcal{I}_A, \mathcal{I}_B) \rightsquigarrow d(A, B) = \left(\sum_{j=1}^p \left(\frac{\delta_j(\mathcal{I}_A^j, \mathcal{I}_B^j)}{m_j} \right)^2 \right)^{\frac{1}{2}}.$$

où m_j est un coefficient de normalisation à définir.

Pour cela, nous appliquerons la propriété suivante :

Propriété :

Soient n individus décrits par les vecteurs d'intervalles $\mathcal{I}_i \in \mathbb{R}^p$ ($i = 1, \dots, n$) et munis des poids p_i de somme égale à 1. Nous pouvons montrer que :

$$\sum_{i=1}^n p_i d^2(\mathcal{I}_i, \bar{\mathcal{I}}) = \sum_{i=1}^n \sum_{i'=1}^n \frac{p_i p_{i'}}{2} d^2(\mathcal{I}_i, \mathcal{I}_{i'})$$

où d est la distance euclidienne et $\bar{\mathcal{I}} = \sum_{i'=1}^n p_{i'} \mathcal{I}_{i'}$.

On peut appliquer cette égalité en prenant les poids $p_i = \frac{1}{n}$. On obtient alors les coefficients de la normalisation par l'inverse de la variance :

$$m_j = \frac{1}{n^2} \sum_{i=1}^n \sum_{i'=1}^n \delta_j^2(\mathcal{I}_i^j, \mathcal{I}_{i'}^j).$$

Une toute autre possibilité de normalisation est la normalisation par l'inverse de l'écart maximum de la variable :

$$m_j = |\max_i \mathcal{I}_i^j - \min_i \mathcal{I}_i^j|$$

où, pour simplifier, m_j est la longueur du domaine de \mathcal{Y}_j .

6.2.2. Fonction de comparaison de Ichino

Cette fonction de comparaison est définie comme suit :

$$\begin{aligned} \delta_j : \mathcal{B}_j \times \mathcal{B}_j &\rightarrow \mathbb{R}^+ \\ (\mathcal{I}_A^j, \mathcal{I}_B^j) &\rightsquigarrow \delta(\mathcal{I}_A^j, \mathcal{I}_B^j) = |\mathcal{I}_A^j \oplus \mathcal{I}_B^j| - |\mathcal{I}_A^j \cap \mathcal{I}_B^j| \\ &\quad + \gamma(2|\mathcal{I}_A^j \cap \mathcal{I}_B^j| - |\mathcal{I}_A^j| - |\mathcal{I}_B^j|) \end{aligned}$$

où $0 \leq \gamma \leq 0.5$ et où $|\mathcal{I}_\bullet^j|$ est l'étendue de l'intervalle \mathcal{I}_\bullet^j . Nous choisissons cependant $\gamma \neq 0$ i.e. $\delta_j(\mathcal{I}_A^j, \mathcal{I}_B^j) = |\mathcal{I}_A^j \oplus \mathcal{I}_B^j| - |\mathcal{I}_A^j \cap \mathcal{I}_B^j|$ car sinon le résultat est identique dans les deux cas suivants :

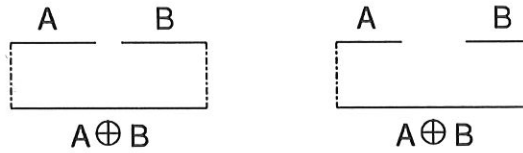


FIGURE 1. Comparaison de deux intervalles

C'est pourquoi nous choisissons plutôt de prendre $\gamma = 0.5$.

Proposition :

Si nous supposons que :

- ◇ $\gamma = 0.5$,
- ◇ $\mathcal{I}_A^j = [a_i^j, a_s^j]$ et $\mathcal{I}_B^j = [b_i^j, b_s^j]$ sont deux intervalles,
- ◇ $\mathcal{I}_A^j \not\subseteq \mathcal{I}_B^j$ et $\mathcal{I}_B^j \not\subseteq \mathcal{I}_A^j$.

alors δ_j mesure la distance entre les milieux a_m^j et b_m^j des intervalles \mathcal{I}_A^j et \mathcal{I}_B^j :

$$\delta_j(\mathcal{I}_A^j, \mathcal{I}_B^j) = |a_m^j - b_m^j|.$$

Démonstration :

Comme $\gamma = 0.5$, on a :

$$\delta_j(\mathcal{I}_A^j, \mathcal{I}_B^j) = |\mathcal{I}_A^j \oplus \mathcal{I}_B^j| - \frac{1}{2}(|\mathcal{I}_A^j| + |\mathcal{I}_B^j|).$$

De plus, $\mathcal{I}_A^j = [a_i^j, a_s^j]$ et $\mathcal{I}_B^j = [b_i^j, b_s^j]$ sont deux intervalles donc, d'après la définition de l'opérateur d'union jointe :

$$\mathcal{I}_A^j \oplus \mathcal{I}_B^j = [\min(a_i^j, b_i^j), \max(a_s^j, b_s^j)] .$$

Comme $\mathcal{I}_A^j \not\subseteq \mathcal{I}_B^j$ et $\mathcal{I}_B^j \not\subseteq \mathcal{I}_A^j$, on a nécessairement :

$$|\mathcal{I}_A^j \oplus \mathcal{I}_B^j| = \max(a_s^j, b_s^j) - \min(a_i^j, b_i^j) = \begin{cases} b_s^j - a_i^j & \text{si } b_s^j \geq a_s^j , \\ a_s^j - b_i^j & \text{si } a_s^j \geq b_s^j . \end{cases}$$

On en déduit que si $b_s^j \geq a_s^j$ alors

$$\begin{aligned} \delta_j(\mathcal{I}_A^j, \mathcal{I}_B^j) &= b_s^j - a_i^j - \frac{1}{2}(a_s^j - a_i^j) - \frac{1}{2}(b_s^j - b_i^j) \\ &= \underbrace{b_s^j - \frac{1}{2}(b_s^j - b_i^j)}_{b_m^j} - \underbrace{[a_i^j + \frac{1}{2}(a_s^j - a_i^j)]}_{a_m^j} . \end{aligned}$$

Même raisonnement si $a_s^j \geq b_s^j$.

□

PARTIE III

MÉTHODES DE CLASSIFICATION

CHAPITRE 7

MÉTHODE HIÉRARCHIQUE DIVISIVE (DIV)

7.1. Généralités sur les méthodes divisives

Les méthodes divisives de classification se basent sur une classification hiérarchique. Partant d'un ensemble Ω d'individus, on construit de manière itérative une famille \mathcal{H} de classes de Ω vérifiant les propriétés suivantes :

- ◇ $\Omega \in \mathcal{H}$,
- ◇ $\forall w \in \Omega : \{w\} \in \mathcal{H}$,
- ◇ $\forall h, h' \in \mathcal{H} : h \cap h' \neq \emptyset \Rightarrow h \subset h' \text{ ou } h' \subset h$.

Cette famille \mathcal{H} est appelée hiérarchie et les classes de \mathcal{H} sont appelées papiers. Elles sont représentées par un arbre appelé arbre hiérarchique (section 5.2.3).

Les méthodes divisives de classification construisent une hiérarchie par divisions successives de Ω jusqu'à la partition des singletons (1 classe $\rightarrow n$ classes). Nous les appellerons méthodes descendantes de classification hiérarchique par opposition aux méthodes ascendantes de classification hiérarchique qui partent de la partition des singletons et qui procèdent par agrégations successives (n classes $\rightarrow 1$ classe).

7.1.1. *Algorithme-type d'une méthode divisive*

La forme générale d'un algorithme divisif de classification pour n individus est :

Initialisation : $P_1 = \Omega$, $k = 1$

Tant que $k < n$ **alors :**

- ◊ Choisir $C \in P_k$
- ◊ Déterminer (C_1, C_2) une partition de C
- ◊ $P_{k+1} = P_k \cup \{C_1, C_2\} - \{C\}$
- ◊ $k \leftarrow k + 1$

Fin Tant que

Cet algorithme nécessite $n - 1$ étapes (en supposant les individus tous différents).

Les différents algorithmes divisifs de classification se distinguent par :

- ◊ le choix de la classe C à diviser.
- ◊ l'algorithme de partitionnement d'une classe C en deux classes C_1 et C_2 .

7.1.2. *Choix de la classe à diviser*

A chaque itération d'un algorithme divisif de classification, on a une partition $P_k = (C_1, C_2, \dots, C_k)$ en k classes de Ω telles que $C_i \cap C_j = \emptyset$ et $\bigcup_{i=1}^k C_i = \Omega$.

Il faut alors choisir la classe à diviser en m sous-classes pour obtenir une partition en $k + m - 1$ classes. Généralement $m = 2$, on obtient une partition en $k + 1$ classes et la hiérarchie peut alors être représentée par un arbre binaire.

La stratégie adoptée par Chavent ⁽¹⁾ consiste à choisir de diviser la classe qui donne la "meilleure" partition au sens d'un critère d'évaluation W . Si

⁽¹⁾Pour plus de détails que donnés ici, nous conseillons de se référer à [6].

$P_k = (C_1, \dots, C_k)$ est une partition en k classes de Ω et que nous divisons une classe C_i en deux classes C_i^1, C_i^2 , on obtiendra une partition $P_{k+1} = (C_1, \dots, C_{i-1}, C_i^1, C_i^2, \dots, C_k)$. Il faut maintenant déterminer comment choisir C_i telle que $W(P_{k+1})$ soit optimal en supposant l'algorithme de partitionnement d'une classe en deux sous-classes choisi. Nous cherchons alors, parmi toutes les partitions en $k + 1$ classes, résultant de la division d'une classe, celle qui optimise le critère. Nous retenons ensuite la classe qui a induit cette partition.

7.1.3. Division de la classe

Une fois choisi notre facteur de division choisi, c'est-à-dire $m = 2$, le problème est alors de trouver une "bonne" partition en deux classes (C_1, C_2) . L'énumération complète de toutes les partitions en deux classes de n individus conduit à évaluer $2^{n-1} - 1$ partitions, ce qui n'est envisageable que pour n très petit. Le critère d'inertie que nous introduirons permettra de contourner ce problème de complexité.

7.1.4. Intérêts et désavantages des méthodes divisives

En général, les ouvrages sur la classification accordent une plus grande importance aux méthodes agglomératives (ascendantes) qu'aux méthodes divisives (descendantes). Nous pouvons expliquer ce désintérêt par la différence de complexité des deux approches. En effet, à la première itération d'une méthode ascendante, il est nécessaire de considérer toutes les agrégations possibles de deux individus parmi n soit $C_n^2 = \frac{n(n-1)}{2}$ possibilités. Un algorithme descendant nécessite quant à lui l'énumération de toutes les divisions de n individus en deux sous-ensembles non vides lors de la première étape, soit $2^{n-1} - 1$ possibilités.

Cependant, le statisticien s'intéresse souvent aux partitions en peu de classes. Ces partitions sont obtenues lors des dernières itérations d'une méthode ascendante, et lors des premières d'une méthode descendante. Le temps de calcul peut donc être moins long et les erreurs pourront moins facilement se propager. De plus, les méthodes divisives fournissent une interprétation plus simple des classes de la hiérarchie que les méthodes ascendantes (surtout dans le cas monothétique).

Une critique commune aux algorithmes hiérarchiques est qu'à chaque étape, nous obtenons la nouvelle partition à partir de la partition obtenue à l'étape précédente sans la remettre en question. Nous ne retrouverons pas ce phénomène propre à ces méthodes lors de l'étude de la méthode des nuées dynamiques.

7.2. Extension du critère d'inertie

Sachant que l'inertie d'une classe C_k s'écrit (Annexe A) :

$$W_k = \sum_{x_i \in C_k} \sum_{x_{i'} \in C_k} \frac{p_i p_{i'}}{2\mu_k} d_M^2(x_i, x_{i'})$$

et en remplaçant la distance euclidienne d_M par une dissimilarité d , nous pouvons définir une extension du critère d'inertie intra-classe au cas d'un tableau de dissimilarité $D = (d_{ii'})$, ($d_{ii'} = d(x_i, x_{i'}) = d(x_{i'}, x_i) \geq 0$, $d_{ii} = 0$) comme suit :

$$W = \sum_{k=1}^m W_k = \sum_{k=1}^m \sum_{x_i \in C_k} \sum_{x_{i'} \in C_k} \frac{p_i p_{i'}}{2\mu_k} d_{ii'}^2.$$

On parlera d'"inertie" intra-classe sur un tableau de dissimilarités.

L'extension de l'inertie totale T du nuage $N(\Omega)$ au cas d'un tableau de dissimilarités $D = (d_{ii'})$ est définie par :

$$T = \sum_{x_i \in \Omega} \sum_{x_{i'} \in \Omega} \frac{p_i p_{i'}}{2\mu_\Omega} d_{ii'}^2.$$

On parlera d'"inertie" totale sur un tableau de dissimilarités.

L'avantage de ce nouveau critère est qu'il est défini sur un tableau de dissimilarités et qu'il peut s'appliquer sur tous les types de données. Son inconvénient est sa complexité, due à la double somme.

Simplifions le critère en utilisant les propriétés des dissimilarités et en fixant les poids :

- ◇ Vu que les dissimilarités sont symétriques et en ordonnant les indices des n_k individus de C_k , on peut écrire :

$$W_k = \sum_{j=1}^{n_k-1} \sum_{i'>i}^{n_k} \frac{p_i p_{i'}}{\mu_k} d_{ii'}^2 .$$

- ◇ Si les n individus de Ω sont munis de poids égaux à 1, nous avons :

$$\mu_k = n_k$$

$$p_i p_{i'} = 1$$

et

$$W_k = \frac{1}{2n_k} \sum_{x_i \in C_k} \sum_{x_{i'} \in C_k} d_{ii'}^2$$

ou encore

$$W_k = \frac{1}{n_k} \sum_{i=1}^{n_k-1} \sum_{i'>i}^{n_k} d_{ii'}^2 .$$

7.3. Choix de la classe à diviser

Lorsqu'il s'agit de données quantitatives (comme les données intervalles), nous choisissons de diviser la classe C dont la partition (C_1, C_2) a une inertie inter-classe maximum (chapitre 3).

Le critère d'évaluation des bi-partitions construites à chaque étape est un critère basé sur l'inertie. L'inertie intra-classe d'une partition se calcule à partir

du centre de gravité des classes. Lorsqu'il s'agit de descriptions symboliques, il est complexe de définir la notion de centre de gravité. Nous avons démontré (Annexe A) que l'inertie d'une classe peut s'écrire comme une double somme pondérée des distances entre les individus deux à deux. Dans la section 7.2, nous avons étendu la définition de l'inertie intra-classe au cas d'un tableau de dissimilarité. Nous utiliserons ici l'extension du critère d'inertie, avec tous les poids p_i égaux à 1 :

$$W_k = \frac{1}{2n_k} \sum_{x_i \in C_k} \sum_{x_{i'} \in C_k} d_{ii'}^2$$

où n_k est le nombre d'individus de la classe C_k .

Nous pouvons alors appliquer la stratégie vue à la section 7.1.2 puisque ce critère est additif. Ainsi, nous divisons la classe C qui maximise :

$$W_C - W_{C_1} - W_{C_2} .$$

L'algorithme divisif de classification hiérarchique de Chavent est donc le suivant :

Initialisation : $P_1 = \Omega$, $k = 1$

Tant que $k < n$ **alors :**

- ◇ Pour chaque classe C de P_k , choisir parmi l'ensemble des partitions (C_1, C_2) de C induites par l'ensemble des questions binaires (section 7.4.1), celle qui minimise :

$$W = \frac{1}{2n_1} \sum_{x_i \in C_1} \sum_{x_{i'} \in C_1} d_{ii'}^2 + \frac{1}{2n_2} \sum_{x_i \in C_2} \sum_{x_{i'} \in C_2} d_{ii'}^2 .$$

- ◇ Choisir $C \in P_k$ qui maximise :

$$W_C - W_{C_1} - W_{C_2}$$

- ◇ $P_{k+1} = P_k \cup \{C_1, C_2\} - \{C\}$

- ◇ $k \leftarrow k + 1$

Fin Tant que

Nous obtenons une hiérarchie dont les singletons sont les n classes de la partition P_n obtenue à la dernière itération de l'algorithme. Les paliers de cette hiérarchie sont indicés par $W_C - W_{C_1} - W_{C_2}$. Les branches de l'arbre sont illustrées par les questions binaires qui ont induit les divisions successives.

7.4. Bipartitionnement d'une classe

Le problème est, rappelons le, de trouver une partition en deux classes (C_1, C_2) d'un ensemble C de n individus, optimale au sens du critère d'évaluation. Pour trouver cette partition optimale par énumération complète, il faudrait évaluer toutes les partitions en deux classes de C , soit $2^n - 1$ partitions.

Nous réduisons le nombre de partitions à évaluer en n'envisageant qu'un sous-ensemble de partitions (C_1, C_2) induites par un ensemble de questions binaires.

7.4.1. Questions binaires et objets symboliques

Une question binaire est une condition à laquelle les individus satisfont ou ne satisfont pas. On peut la représenter de la manière suivante :

$$[Y_i \in V] \text{ ou } [Y_i \in \bar{V}]$$

où (V, \bar{V}) est une partition en deux classes du domaine d'observation \mathcal{Y}_i de la variable Y_i telle que :

$$V \cap \bar{V} = \emptyset \text{ et } V \cup \bar{V} = \mathcal{Y}_i .$$

Lorsque le domaine d'observation \mathcal{Y}_i est ordonné (c'est le cas pour les intervalles), la partition (V, \bar{V}) peut être entièrement définie par une valeur $c \in \mathcal{Y}_i$ appelée valeur de coupure. La question binaire est alors représentée comme

suit :

$$[Y_i \leq c] \text{ ou } [Y_i > c] .$$

Une question binaire s'identifie à une fonction binaire Q_V définie de Ω dans $\{0, 1\}$. Nous notons 1 la réponse positive à $[Y_i \in V]$ et inversement la réponse négative par 0. Une partition (C_1, C_2) est induite par cette question binaire comme suit :

$$C_1 = \{x \in C \mid Q_V(x) = 1\} \text{ et } C_2 = \{x \in C \mid Q_V(x) = 0\} .$$

Dans le cas particulier où les individus de C ont une description univaluée, la fonction binaire Q_V est définie par :

$$Q_V(x) = \begin{cases} 1 & \text{si } Y_i(x) \in V \\ 0 & \text{si } Y_i(x) \in \bar{V} . \end{cases}$$

Dans le cas où les individus ont une description de type intervalle, nous ne pouvons plus répondre directement à la question : $[Y_i \in V]!$

En effet, dans le cas où les individus sont décrits par des intervalles, la description $Y_i(x)$ d'un individu x sur Y_i n'appartient plus nécessairement à V ou à \bar{V} (ex : taille d'un individu décrite par l'intervalle $[170, 175]$, on ne peut dire si sa taille est plus grande ou plus petite que 172). En revanche, en considérant que sa taille est uniformément distribuée sur $[170, 175]$ (nous ferons toujours cette hypothèse), nous connaissons les probabilités qu'elle soit inférieure ($\frac{2}{5}$) et supérieure ($\frac{3}{5}$) à 172. En fonction de ces probabilités, nous choisirons de répondre à la question.

Nous répondons donc à la question binaire

$$[Y_i \in V] \text{ ou } [Y_i \in \bar{V}]$$

par l'intermédiaire de deux objets symboliques assertions s_1 et s_2 de descriptions respectives V et \bar{V} et de fonctions d'appartenance a_1 et a_2 définies de Ω

dans $[0, 1]$. Les degrés d'appartenance de $a_1(x)$ et $a_2(x)$ s'interprètent comme les degrés de vérité des propriétés, $[Y_i(x) \in V]$ et $[Y_i(x) \in \bar{V}]$. Ainsi, nous aurons toujours

$$a_1(x) + a_2(x) = 1 .$$

La fonction Q_V associée à la question $[Y_i \in V]$ est alors définie par :

$$Q_V(x) = \begin{cases} 1 & \text{si } a_1(x) \geq \frac{1}{2} \\ 0 & \text{si } a_1(x) < \frac{1}{2} . \end{cases}$$

Dans le cas particulier des données intervalles, nous verrons que la fonction Q_c associée à la question $[Y_i \leq c]$ peut être définie par :

$$Q_c(x) = \begin{cases} 1 & \text{si } m_x \leq c \\ 0 & \text{si } m_x > c \end{cases} .$$

où m_x est calculée à partir de la description $Y_i(x) = f_x$. Nous appellerons m_x la valeur "médiane" de la description de x sur Y_i .

La valeur médiane m_x d'un individu x sur une variable Y_i est définie de la manière suivante :

$$\int_{-\infty}^{m_x} f_x(y) dy = \frac{1}{2}$$

Si f_x est une distribution de probabilité uniforme sur $[a, b]$ (hypothèse que nous ferons), alors m_x est le milieu de $[a, b]$.

Nous pouvons aussi définir, sachant c (la valeur de coupure) :

$$a_1(x) = \int_{-\infty}^c f_x(y) dy$$

$$a_2(x) = \int_c^{\infty} f_x(y) dy = 1 - a_1(x)$$

7.4.2. L'ensemble des questions binaires

Nous allons voir comment définir l'ensemble des questions binaires sur une variable continue, c'est-à-dire l'ensemble des valeurs de coupure c .

Quand une variable Y_i est continue (comme c'est le cas pour un intervalle), il existe une infinité de valeurs de coupure $c \in \mathcal{Y}_i \subset \mathbb{R}$ et donc une infinité de questions binaires. En revanche, il y a un nombre fini de bi-partitions induites. Si le mode de description des n individus sur Y_i est un intervalle, nous avons que $Y_i(x)$ est une distribution uniforme sur \mathcal{Y}_i , et pour chacune de ces n distributions, nous pouvons déterminer leurs valeurs "médianes" (les milieux des intervalles). Après les avoir ordonnées, nous notons ces n valeurs m_1, \dots, m_n .

Nous en déduisons qu'il y a au plus $n - 1$ partitions de (C_1, C_2) induites par les questions binaires sur Y_i car quelle que soit la valeur de coupure c entre deux valeurs consécutives m_j et m_{j+1} , la partition induite est la même.

En conclusion, nous pouvons calculer facilement le nombre de bi-partitions à évaluer de la manière suivante : si z_1 est le nombre de partitions (C_1, C_2) induites par l'ensemble des questions binaires sur Y_i , il faudra évaluer au total $z_1 + \dots + z_p$ partitions (C_1, C_2) .

7.4.3. Choix d'une question binaire

Supposons maintenant que nous ayons évalué les $z_1 + \dots + z_p$ bi-partitions, et que nous ayons retenu la bi-partition (C_1, C_2) , la meilleure au sens de notre critère d'évaluation. Cette bi-partition pouvait être induite par plusieurs questions binaires, le problème reste de choisir une de ces questions.

Parmi les questions définies sur une même variable Y_i induisant (C_1, C_2) , nous choisirons celle qui "discrimine" au mieux la partition (C_1, C_2) . Nous définissons, dans ce but, un critère à partir des fonctions d'appartenance a_1 et a_2

(section 7.4.1), que nous essayerons de minimiser. Nous noterons E' l'ensemble de questions binaires qui minimisent ce critère.

Soit E l'ensemble des questions $[Y_i \in V]$ ou encore, l'ensemble des partitions (V, \bar{V}) qui induisent la partition (C_1, C_2) , on veut maintenant choisir $(V, \bar{V}) \in E$ qui "discrimine" le plus les individus de C_1 et de C_2 . Par exemple, si $x \in C_1$, on veut que $a_1(x)$ soit maximale tandis que $a_2(x)$ est minimale. Au mieux, nous trouvons $a_1(x) = 1$ et $a_2(x) = 0$ car dans ce cas, x est parfaitement "discriminé".

Pour choisir cette coupure, nous allons chercher à minimiser sur E le critère :

$$R = \sum_{x \in C_1} (a_2(x))^2 + \sum_{x \in C_2} (a_1(x))^2$$

tel que si les individus de C_1 et de C_2 sont parfaitement discriminés, $R = 0$ ($\forall x \in C_1, a_2(x) = 0$ et $\forall x \in C_2, a_1(x) = 0$).

L'ensemble E des partitions (V, \bar{V}) qui induisent C_1 et C_2 correspond aux valeurs de coupure c , appartenant à l'intervalle $[m, m']$ des deux "médianes" ayant déterminé la partition (section 7.4.1). En effet, tous les individus qui ont une "médiane" inférieure ou égale à m ont été affectés à C_1 les autres à C_2 . Puisque nous connaissons cette partition (C_1, C_2) , nous connaissons l'ensemble E des partitions retenues que nous notons $E = [m, m']$.

Dans notre cas, nous pouvons donc réécrire le critère R en fonction des définitions des fonctions $a_1(x)$ et $a_2(x)$ que nous avons données :

$$R(c) = \sum_{x \in C_1} \left(\int_c^\infty f_x(y) dy \right)^2 + \sum_{x \in C_2} \left(\int_{-\infty}^c f_x(y) dy \right)^2.$$

R est alors une fonction de c que nous allons chercher à minimiser sur $[m, m']$.

Admettons, sans démonstration, le théorème suivant :

Théorème :

Supposons que :

- ◇ la partition (C_1, C_2) est connue et engendrée par les coupures $c \in [m, m']$,
- ◇ f_x est une distribution uniforme sur \mathbb{R} .

Alors, la fonction $R(c)$ est convexe et $\exists [c_1, c_2] \subset [m, m']$ tel que :

$$\forall c \in [c_1, c_2], R(c) = \min_{c \in [m, m']} R(c)$$

Dans le cas des intervalles, les fonctions f_x sont toutes des distributions uniformes sur un intervalle $[a, b]$. Les hypothèses du théorème sont donc vérifiées et nous allons chercher la valeur de coupure c .

Notons $[a, b]$ les intervalles décrivant les individus de C_1 et $[a', b']$ les intervalles décrivant les individus de C_2 . Posons ensuite :

$$\begin{aligned} \alpha &= \max_{x \in C_1} b, \\ \beta &= \min_{x \in C_2} a'. \end{aligned}$$

- ◇ Si $\alpha \leq \beta$, nous avons de suite :

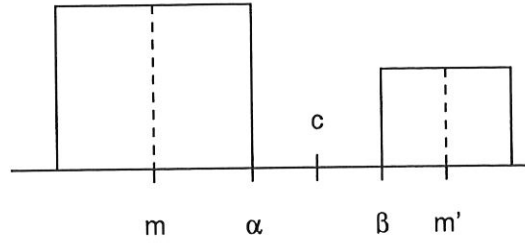
$$[c_1, c_2] = [\alpha, \beta].$$

En effet, $\forall c \in [\alpha, \beta]$, on a :

$$R(c) = \sum_{x \in C_1} \left(\int_c^\infty f_x(y) dy \right)^2 + \sum_{x \in C_2} \left(\int_{-\infty}^c f_x(y) dy \right)^2 = 0.$$

On choisit dans $[\alpha, \beta]$ la valeur de coupure :

$$c = \frac{\alpha + \beta}{2}.$$

FIGURE 1. Coupure c si $\alpha \leq \beta$

◇ Si $\beta \leq \alpha$, nous montrons que :

$$[c_1, c_2] \subset [\max(m, \beta), \min(m', \alpha)]$$

En effet, la fonction

$$\sum_{x \in C_1} \left(\int_c^\infty f_x(y) dy \right)^2$$

décroît strictement en fonction de c sur $[m, \beta]$.

Par ailleurs, $\forall c \leq \beta$:

$$\sum_{x \in C_2} \left(\int_{-\infty}^c f_x(y) dy \right)^2 = 0$$

On en déduit que $R(c)$ est strictement décroissante sur $[m, \beta]$.

De même, si $\alpha \leq m'$ alors $\forall c \in [\alpha, m']$ on a :

$$\sum_{x \in C_1} \left(\int_c^\infty f_x(y) dy \right)^2 = 0$$

et

$$\sum_{x \in C_2} \left(\int_{-\infty}^c f_x(y) dy \right)^2$$

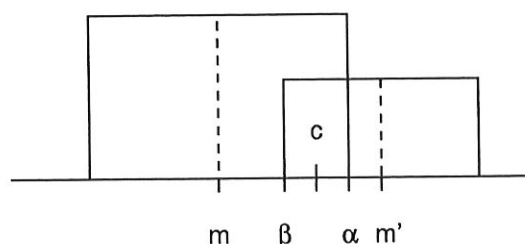
croît strictement en fonction de c sur $[\alpha, m']$.

On en déduit que $R(c)$ est strictement croissante sur $[\alpha, m']$.

On en déduit :

$$[c_1, c_2] \subset [\max(m, \beta), \min(m', \alpha)]$$

Même si nous ne savons pas expliciter simplement la solution $[c_1, c_2]$,

FIGURE 2. Coupure c si $\beta \leq \alpha$

$[\max(m, \beta), \min(m', \alpha)]$ est un encadrement plus précis que $[m, m']$ de l'ensemble des solutions.

On choisit donc :

$$c = \frac{\max(m, \beta) + \min(m', \alpha)}{2}$$

7.5. Complexité

La complexité de cette méthode se situe principalement :

- ◇ au niveau du nombre de bi-partitions à évaluer sur chaque variable. A partir des résultats vus à la section 7.4.2 à propos du nombre de bi-partitions à évaluer, on établit que la complexité de cette partie est en $\mathcal{O}(n)$
- ◇ au niveau du calcul du critère d'évaluation des bi-partitions. Le calcul du critère d'"inertie" intra-classe (section 7.2) sur tableau de dissimilarités a une complexité en $\mathcal{O}(n^2)$. Lorsque le domaine d'observation de la variable de coupure est ordonné, on peut proposer une mise à jour du critère réduisant la complexité à $\mathcal{O}(n)$.

7.6. Algorithmes

7.6.1. Mise à jour du critère d'inertie

Nous avons vu que lorsque le domaine d'observation d'une variable Y_i est ordonné, cette variable définit un ordre sur les n individus d'une classe C . Dans le cas des intervalles, on ordonne les individus selon leurs "médianes" sur Y_i :

on note m_1, \dots, m_n ces n valeurs ordonnées. L'algorithme de partitionnement en deux classes peut maintenant évaluer les bi-partitions respectant cet ordre plus rapidement car il n'y en a qu'au plus $n - 1$ différentes.

Soient (x_1, \dots, x_n) les n individus ordonnés, (C_1, C_2) la bi-partition constituée des n_1 individus d'une part et des $n - n_1$ derniers d'autre part, (C'_1, C'_2) la bi-partition obtenue en affectant le premier individu x_{n_1+1} de C_2 à C_1 et d la dissimilarité choisie (Rappel : $d(x_i, x_{i'})$ est noté $d_{i,i'}$).

En utilisant la dernière formulation de l'"inertie" d'une classe vue à la section 7.2, et ayant défini W (resp. W') comme étant l'"inertie" intra-classe de la bipartition (C_1, C_2) (resp. (C'_1, C'_2)), nous avons par [6] :

$$W = \frac{1}{n_1} \underbrace{\sum_{i=1}^{n_1-1} \sum_{i'>i}^{n_1} d_{ii'}^2}_A + \frac{1}{n - n_1} \underbrace{\sum_{i=n_1+1}^{n-1} \sum_{i'>i}^n d_{ii'}^2}_B$$

et si on affecte le $n_1 + 1^e$ individu à C_1 , nous avons :

$$W' = \frac{1}{n_1 + 1} \underbrace{\sum_{i=1}^{n_1} \sum_{i'>i}^{n_1+1} d_{ii'}^2}_{A'} + \frac{1}{n - n_1 - 1} \underbrace{\sum_{i=n_1+2}^{n-1} \sum_{i'>i}^n d_{ii'}^2}_{B'}.$$

De plus, nous avons que :

$$A' = \underbrace{\sum_{i=1}^{n_1-1} \sum_{i'>i}^{n_1} d_{ii'}^2}_A + \sum_{i=1}^{n_1} d_{i,n_1+1}^2$$

et

$$B' = \underbrace{\sum_{i=n_1+1}^{n-1} \sum_{i'>i}^n d_{ii'}^2}_B - \sum_{i>n_1+1}^n d_{i,n_1+1}^2.$$

On en déduit W' en fonction de A et B :

$$W' = \frac{1}{n_1 + 1} \left(A + \sum_{i=1}^{n_1} d_{i,n_1+1}^2 \right) + \frac{1}{n - n_1 - 1} \left(B - \sum_{i>n_1+1}^n d_{i,n_1+1}^2 \right)$$

Si le calcul initial ($n_1 = 0$) de A, B et W est de complexité $\mathcal{O}(n^2)$, la mise à jour de A, B et W est ensuite de complexité $\mathcal{O}(n)$.

7.6.2. Choix de la meilleure coupure d'une classe

Soit C une classe de n individus. On veut trouver la coupure qui induit la bi-partition (C_1, C_2) de plus petite "inertie" intra-classe W . Pour cela, on recherche sur chaque variable Y_i la valeur qui induit la meilleure bi-partition et on retient ensuite la variable correspondante.

Soit (x_1, \dots, x_n) , la liste des n individus de C ordonnés en fonction de leurs "médianes" sur Y_i et soit (m_1, \dots, m_n) la liste de ces n "médianes" ordonnées. Nous exécutons l'algorithme suivant qui permet de trouver la valeur V^* qui induit la bi-partition de plus petite "inertie" intra-classe W^* :

Initialisation :

◇ Trouver la première valeur m_k telle que $m_k \neq m_{k+1}$.

◇ Calcul des valeurs intermédiaires :

$$A = \sum_{i=1}^{k-1} \sum_{i' > i}^k d_{ii'}^2$$

$$B = \sum_{i=k+1}^{n-1} \sum_{i' > i}^n d_{ii'}^2$$

$$W = \frac{1}{k} A + \frac{1}{n-k} B$$

◇ Calcul de l'inertie et de la valeur de coupure :

$$W^* = W$$

$$V^* = \frac{m_k + m_{k+1}}{2}$$

Tant que $k < n - 1$ **alors** :

◇ Calculer :

$$A = A + \sum_{j=1}^k d_{j,k+1}^2$$

$$B = B - \sum_{j=k2+}^n d_{j,k+1}^2$$

$$W = \frac{1}{k+1} A + \frac{1}{n-k-1} B$$

- ◇ $k \leftarrow k + 1$
- ◇ Si $W \leq W^*$ et $m_k \neq m_{k+1}$ alors :

$$W^* = W$$

$$V^* = \frac{m_k + m_{k+1} + 1}{2}$$

Fin Tant que

7.6.3. Algorithme principal

L'algorithme peut être défini à partir des deux listes suivantes :

- ◇ T est la liste des noeuds de l'arbre
- ◇ F est la liste des feuilles de l'arbre

Chaque noeud et chaque feuille est composé d'une bi-partition, d'une question binaire et d'un indice valant 0 si la feuille est réduite à un singleton. On possède également en input :

- ◇ le tableau $D(.,.)_{n \times n}$ des dissimilarités entre les n individus de Ω .
- ◇ la matrice des données symboliques $\tilde{X}_{n \times p}$ où p est le nombre de variables.
- ◇ $nbfeuilles_{\max}$ le nombre maximum de feuilles dans F ($nbfeuilles_{\max} \leq n$ et $nbfeuilles_{\max} = n$ uniquement si tous les individus prennent des valeurs différentes).

Une question binaire $[Y \in V]$ sera représentée ici par le couple (Y, V) appelé coupure. L'"inertie" intra-classe sera quant à elle notée W .

Initialisation

- ◇ Pour i allant de 1 à p faire
 - Pour j allant de 1 à n faire
 - Calculer la médiane m_j^i de l'élément $\tilde{X}(j, i)$

◇ Pour i allant de 1 à p faire

Trier pour j allant de 1 à n les médianes m_j^i

On se servira de cet ordre pour mettre à jour W plus facilement (section 7.6.1)

◇ Trouver la coupure (Y, V) et la bi-partition (C_1, C_2) de Ω maximisant le critère $\Delta\Omega = W_\Omega - W_{C_1} - W_{C_2}$ (section 7.3)

$f = ((C_1, C_2), (Y, V), \Delta(\Omega))$

◇ $T \leftarrow \emptyset; F \leftarrow \{f\}; k \leftarrow 1$

Tant que $k \leq nbfeuilles_{\max} - 1$ **alors :**

◇ Choisir dans F la feuille (non-réduite à un singleton) $f = ((C_1, C_2), (Y, V), \Delta(C))$ qui maximise $\Delta(C)$

◇ **Si** $k + 1 \neq nbfeuilles_{\max} - 1$ **alors :**

Si $W_{C_1} \neq 0$ (les individus de C_1 ne sont pas tous identiques) **alors :**

Trouver la coupure (Y_1, V_1) et la bi-partition (C_1^1, C_1^2) de C_1

Calculer $\Delta(C_1)$

$f_1 = ((C_1^1, C_1^2), (Y_1, V_1), \Delta(C_1))$

Sinon $f_1 = (C_1, 0)$

Si $W_{C_2} \neq 0$ **alors :**

Trouver la coupure (Y_2, V_2) et la bi-partition (C_2^1, C_2^2) de C_2

Calculer $\Delta(C_2)$

$f_2 = ((C_2^1, C_2^2), (Y_2, V_2), \Delta(C_2))$

Sinon $f_2 = (C_2, 0)$

Sinon

$f_1 = (C_1, 0)$

$f_2 = (C_2, 0)$

◇ $T \leftarrow T \cup \{f\}; F \leftarrow F \cup \{f_1, f_2\}; k \leftarrow k + 1$

Fin Tant que

CHAPITRE 8

MÉTHODE DES NUÉES DYNAMIQUES (DYCLASS)

8.1. Introduction

La méthode des nuées dynamiques de Diday est une méthode d'optimisation d'un critère mathématique. Le problème est de rechercher simultanément la classification et sa représentation qui optimisent ce critère. Cette technique de partitionnement aura pour but de fournir une partition en m classes d'individus (m fixé). L'algorithme nécessite pour cela la définition d'un mode de représentation de tout groupe d'individus. Ce représentant appelé noyau peut être une droite, un centre de gravité, un groupe de points. Nos m noyaux de départ choisis au hasard parmi les noyaux admissibles, on leur affecte les points de la population dont ils sont les plus "proches". On obtient une partition dont on calcule les noyaux et itérativement, on souhaite converger vers une solution minimisant notre critère mathématique. Cet algorithme fait donc décroître un critère W qui mesure la dissimilarité entre les individus des classes et leurs noyaux respectifs. Ce critère est donc une application du type :

$$W : \mathcal{L}_m \times \mathcal{C}_m \rightarrow \mathbb{R}^+$$

où

- ◊ $\mathcal{L}_m = \Omega^m$ est l'ensemble des m -uples noyaux $L = (L_1, \dots, L_m)$ avec $L_k \in \Omega$,
- ◊ \mathcal{C}_m est l'ensemble des partitions $C = (C_1, \dots, C_m)$ à m classes de Ω ,

avec $L = (L_1, \dots, L_m)$ où L_k est l'élément de \mathcal{L} qui minimise $\sum_{x \in C_k} d(x, \cdot)$.

La fonction d'affectation f est définie comme suit :

$$\begin{aligned} f : \mathcal{L}^k &\rightarrow \mathcal{C}_k \\ L &\rightsquigarrow C = f(L) \end{aligned}$$

avec $C = (C_1, \dots, C_m)$ où $C_k = \{x \in \Omega \mid d(x, L_k) \geq d(x, L_l) \forall l = 1, \dots, m \text{ avec } k < l \text{ en cas d'égalité}\}$;

- ◇ A partir d'une solution (C^0, L^0) estimée ou tirée au hasard, nous allons optimiser le critère W par itérations successives en L (C fixé) en utilisant la fonction de représentation g , puis en C (L fixé) en utilisant la fonction d'affectation f jusqu'à obtenir une solution stable. L'algorithme peut se formaliser à l'aide des deux suites v_n et u_n suivantes :

- $v_0 = (C^0, L^0)$ et $v_{n+1} = (C^{n+1}, L^{n+1})$ où $C^{n+1} = f(L^n)$ et $L^{n+1} = g(C^{n+1})$,
- $u_n = W(v_n)$;

- ◇ A partir de v_0 , on calcule v_n jusqu'à sa convergence, on montre alors que cette suite finie converge si elle est stationnaire ($\exists N \mid \forall n > N : v_n = v_{n+1}$).

Proposition :

La suite u_n converge en décroissant.

Démonstration :

Nous devons montrer que $u_n \geq W(C^n, L^{n+1}) \geq u_{n+1}$.

La première inégalité est vraie car $\forall k$ ($k = 1, \dots, m$) : $d(C_k^n, L_k^n) \geq d(C_k^n, L_k^{n+1})$ par définition de g , d'où en sommant sur k :

$$u_n = W(C^n, L^n) \geq W(C^n, L^{n+1}) .$$

La seconde inégalité s'écrit :

$$\sum_{k=1}^m \sum_{x \in C_k^n} d(x, L_k^{n+1}) \geq \sum_{k=1}^m \sum_{x \in C_k^{n+1}} d(x, L_k^{n+1}) .$$

Elle est également vraie par définition de f . La suite ne pouvant prendre qu'un nombre fini de valeurs positives, elle converge en un nombre fini d'itérations et est stationnaire.

□

Proposition :

La suite v_n est stationnaire.

Démonstration :

Supposons que la stationnarité de u_n est atteinte à l'itération N alors $u_{N+1} = u_N$ et $W(v_{N+1}) = W(v_N)$. Cette égalité n'entraîne pas nécessairement que $v_{N+1} = v_N$ car W n'est pas nécessairement injective. Cependant, on peut montrer que $v_n = v_N \forall n \geq N$.

Vu que $u_{N+1} = u_N$, nous avons :

$$W(C^N, L^N) = W(C^N, L^{N+1}) = W(C^{N+1}, L^{N+1}) .$$

L'élément L minimisant $\sum_{x \in C^N} d(x, .)$ étant unique par définition, nous avons que $L^N = L^{N+1}$. Or, $C^N = f(L^N)$ et $C^{N+1} = f(L^{N+1})$ donc $C^N = C^{N+1}$. Par récurrence, on en déduit la stationnarité de la suite v_n .

□

8.4. Le cas des intervalles

8.4.1. L'espace de représentation

L'espace de représentation spécifique sera ici celui des données intervalles à p dimensions où p est le nombre de variables. Les objets sont donc des hyperrectangles dans l'espace euclidien R^p ⁽¹⁾.

Pour appliquer la méthode des nuées dynamiques aux données intervalles, nous devons également définir la mesure de ressemblance entre individus : on choisira la mesure de Hausdorff à p dimensions (indice de dissimilarité). Par la formule de composition vue à la section 6.2.1 (avec $\alpha = 1$), nous pouvons définir :

$$d(\mathcal{I}_k, \mathcal{I}_l) = \sum_{j=1}^p \max\{|a_k^j - a_l^j|, |b_k^j - b_l^j|\}$$

représentant la distance entre les deux vecteurs d'intervalles $\mathcal{I}_k = [a_k, b_k]$ et $\mathcal{I}_l = [a_l, b_l]$ se décomposant selon les variables de la manière suivante : $\mathcal{I}_k^j = [a_k^j, b_k^j]$ et $\mathcal{I}_l^j = [a_l^j, b_l^j]$, ($j = 1, \dots, p$).

8.4.2. Le critère de minimisation

Pour pouvoir calculer le critère de minimisation W , nous devons être capable de calculer l'adéquation entre les individus d'une classe et leur noyau. Cependant, comment calculer le centroïde d'une classe d'intervalles. Le problème général d'optimisation est donc de trouver le centroïde L d'une classe $C = \{\mathcal{I}_i \in \mathcal{I}^p \mid i = 1, \dots, n\}$ ($\mathcal{I} = \{[a, b] \mid a, b \in \mathbb{R}, -\infty < a < b < \infty\}$, n est le nombre d'individu de la classe C) tel que :

$$L = \arg \min_{\mathcal{I} \in \mathcal{I}^p} \sum_{i=1}^n d(\mathcal{I}_i, \mathcal{I}) .$$

⁽¹⁾La méthode des nuées dynamiques a été adaptée aux données symboliques (dont les données intervalles) par Lechevallier.

Par additivité de la mesure de Hausdorff :

$$f(L) = \sum_{j=1}^p \sum_{i=1}^n d(\mathcal{I}_i^j, L^j) .$$

Le problème général d'optimisation peut être réduit à p problèmes locaux
 $f(y^j) = \sum_{i=1}^n d(\mathcal{I}_i^j, L^j)$.

Soit $C = \{\mathcal{I}_i = [a_i, b_i] \mid i = 1, \dots, n\}$, un ensemble de n intervalles de \mathcal{I} .
 Nous devons maintenant trouver $L = [a, b] \in \mathcal{I}$ qui minimise :

$$f(L) = \sum_{i=1}^n d(\mathcal{I}_i, L) = \sum_{i=1}^n \max\{|a - a_i|, |b - b_i|\} .$$

Soit m le milieu de l'intervalle $[a, b]$ et l sa demi-longueur :

$$\begin{aligned} m &= \frac{a+b}{2} \\ l &= \frac{b-a}{2} \\ a &= m-l \\ b &= m+l \end{aligned}$$

$$\begin{aligned} f(L) &= \sum_{i=1}^n \max\{|(m-l) - (m_i - l_i)|, |(m+l) - (m_i + l_i)|\} \\ &= \sum_{i=1}^n \max\{|(m - m_i) - (l - l_i)|, |(m - m_i) + (l - l_i)|\} \\ &= \sum_{i=1}^n (|m - m_i| + |l - l_i|) \\ &= \sum_{i=1}^n |m - m_i| + \sum_{i=1}^n |l - l_i| \end{aligned}$$

par la propriété définie pour $x, y \in \mathbb{R}$: $\max\{|x - y|, |x + y|\} = |x| + |y|$.

Le problème est donc réduit à deux minimisations simples L_1 . Les solutions m et l sont respectivement la médiane de l'ensemble des milieux des n intervalles de C et la médiane de l'ensemble des demi-longueurs (les médianes sont des intervalles si le nombre d'objets est pair, on prendra les milieux de ces intervalles). En conclusion, $L = [a, b] = [m - l, m + l]$.

8.4.3. Construction de l'algorithme

Considérons le partitionnement de l'ensemble $\Omega = \{1, \dots, n\}$ de n individus en m classes disjointes dans le cas où chaque individu est décrit selon chaque variable j ($j = 1, \dots, p$) par un intervalle $\in \mathcal{I}$. L'algorithme des nuées dynamiques va chercher la partition $C = (C_1, \dots, C_m)$ de Ω et l'ensemble $L = (L_1, \dots, L_m)$ de noyaux optimisant le critère de partitionnement basé sur la distance de Hausdorff à p -dimensions :

$$\begin{aligned} W(C, L) &= \sum_{k=1}^m \sum_{i \in C_k} d(\mathcal{I}_i, L_k) \\ &= \sum_{k=1}^m \sum_{i \in C_k} \sum_{j=1}^p \max\{|a_i^j - a_k^j|, |b_i^j - b_k^j|\} \end{aligned}$$

Durant l'étape de représentation, l'algorithme calcule pour chaque classe C_k , le centroïde L_k qui minimise le critère suivant :

$$\sum_{i=1}^n d(\mathcal{I}_i, L) = \sum_{i=1}^n \sum_{j=1}^p d(\mathcal{I}_i^j, L^j)$$

Les centroïdes optimaux sont, comme nous l'avons vu, selon chaque variable j , les intervalles $[a_k^j, b_k^j] = [m_k^j - l_k^j, m_k^j + l_k^j]$.

Durant l'étape d'affectation, l'algorithme calcule une nouvelle partition en affectant chaque objet i au plus proche centroïde L_{k*} où :

$$k* = \arg \min_{k=1, \dots, m} d(\mathcal{I}_i, L_k)$$

8.5. Algorithme MND pour les données intervalles**Initialisation :**Choisir m objets L_1, \dots, L_m parmi l'ensemble des données initiales Ω .**Test :** $Test \leftarrow 0$ **Etape d'affectation :**Pour i allant de 1 à n fairecalculer k^* tel que $k^* = \arg \min_{k=1, \dots, m} d(\mathcal{I}_i, L_k)$ **Si** $i \in C_k$ **et** $k^* \neq k$ **alors :** $Test \leftarrow 1$ $C_{k^*} \leftarrow C_{k^*} \cup \{i\}$ $C_k \leftarrow C_k \setminus \{i\}$ **Etape de représentation :**Pour k allant de 1 à m fairecalculer les centroïdes $L_k = [m_k - l_k, m_k + l_k]$: $m_k^j = \text{médiane}\{m_i^j \mid i \in C_k\}$ $l_k^j = \text{médiane}\{l_i^j \mid i \in C_k\}$ **Arrêt :****Si** $Test = 0$ **alors :**

Sortie

Sinon goto Test :

PARTIE IV

PROGRAMMES ET RÉSULTATS

CHAPITRE 9

INTERPRÉTATION D'UNE PARTITION

Les deux méthodes vues peuvent paraître assez différentes. En effet, la méthode divisive de Chavent part d'une partition unique et découpe cette partition jusqu'à obtenir le nombre de classes voulu. Par contre, pour la méthode des nuées dynamiques, on fixe le nombre de classes, on choisit des représentants, on calcule la partition associée, puis on ajuste cette partition par itérations successives. Cependant, ces deux méthodes ont un point commun important : elles se basent sur un critère inertiel. Ce critère inertiel est explicite pour la méthode de Chavent : maximiser $W_C - W_{C_1} - W_{C_2} = W_C - (W_{C_1} + W_{C_2})$ (section 7.3), on essaie donc de minimiser l'inertie intra-classe de la nouvelle partition. Pour la méthode des nuées dynamiques, on essaie à chaque itération de calculer de meilleurs noyaux afin de minimiser la dissimilarité entre celui-ci et les éléments de la classe qu'il définit, on essaie donc également de minimiser l'inertie intra-classe à chaque itération.

9.1. Inertie expliquée d'une partition

Dès lors, nous sommes amenés à définir un indice expliquant la qualité d'une partition : le pourcentage d'inertie expliquée d'une partition C d'un ensemble Ω de données est égal à :

$$R = \frac{B}{T} \times 100 = \left(1 - \frac{W}{T}\right) \times 100$$

où : T est l'inertie totale du nuage de points, B est son inertie inter-classe et W est son inertie intra-classe. Ces inerties peuvent être également calculées à partir des tableaux de dissimilarités comme on l'a vu à la section 7.2 dans le cas de variables intervalles. Pour la méthode de Chavent, l'inertie expliquée augmente à chaque itération et vaut 100% pour la partition des singletons tandis que pour la méthode des nuées dynamiques, si le nombre de classes désiré est plus petit que le nombre d'objets à classer, on n'atteindra jamais 100%.

9.2. Comparaison de partitions

9.2.1. Le tableau de contingence

La première étape sera de construire un tableau de contingence. Les partitions P_1 et P_2 en m classes des n individus définissent deux variables qualitatives Y_1 et Y_2 à m modalités. Si l'individu j appartient à la classe C_k (resp. $C_{k'}$) de P_1 (resp. P_2) alors $Y_1(x_j) = k$ (resp. $Y_2(x_j) = k'$). On construit alors le tableau de contingence à partir de $n_{kk'}$, le nombre d'individus appartenant à la classe C_k de P_1 et à la classe $C_{k'}$ de P_2 .

9.2.2. Le χ^2 d'indépendance

Une première idée de comparaison est d'effectuer le test du χ^2 d'indépendance :

$$\chi^2(Y_1, Y_2) = \sum_k \sum_{k'} \frac{(n_{kk'} - \frac{n}{m^2})^2}{\frac{n}{m^2}}$$

où les fréquences observées sont les $n_{kk'}$ du tableau de contingence et la fréquence théorique est la répartition moyenne des individus dans le tableau de contingence ($\frac{n}{m^2}$). La zone de rejet du test est calculée en tenant compte du degré de liberté qui sera égal dans notre cas à $(m-1) \times (m-1)$ où m est le nombre de classes. Le test sera rejeté (donc non-indépendance) pour de grandes valeurs. Ce test est implémenté dans le programme `comparaison.cpp` (Annexe B) que nous avons rédigé.

9.2.3. Maximisation "ligne-colonne" du tableau de contingence

Afin d'essayer de faire correspondre les classes issues des partitionnements, il ne suffit pas de repérer simplement les fréquences observées $n_{kk'}$ les plus importantes. Pour être sur d'avoir une correspondance unique entre les classes selon les deux méthodes, nous ne pouvons prendre qu'un élément du tableau selon chaque ligne et colonne.

Nous choisissons donc la fréquence observée la plus élevée dans le tableau et annulons les autres valeurs se trouvant sur sa ligne et sa colonne. Nous répétons cette opération m fois et sommons les valeurs maximales obtenues à chaque itération. Nous obtenons alors un nombre d'individus "bien classés" selon les méthodes et nous pouvons extraire un pourcentage d'individus "bien classés". Cette idée est implémentée dans le programme `comparaison.cpp` (Annexe B).

CHAPITRE 10

APPLICATION : DONNÉES RÉELLES

10.1. Jeu de données "temp1988.sds"

Le fichier de données consiste en la récolte des plages de températures de 60 villes chinoises par mois en 1988. Cela correspond donc à 12 variables intervalles observées sur 60 individus.

10.1.1. La méthode divisive de Chavent

Nous appliquons la méthode de Chavent au jeu de données⁽¹⁾ en faisant les choix suivants : une division jusqu'en 5 classes et une non-normalisation de la distance puisque les variables sont de même type (toutes des mesures de températures). Nous obtenons les résultats suivants, extraits du fichier temp1988.lst :

PARTITION IN 2 CLUSTERS :

Cluster 1 (Cardinal=21) : [October <= 13.200000]			
"BoKeTu_1988"	"ChangChun_1988"	"HaErBin_1988"	"Hailaer_1988"
"HaMi_1988"	"HuHeHaoTe_1988"	"Jiu Quan_1988"	"LanZhou_1988"
"LaSa_1988"	"HuDanJiang_1988"	"NenJiang_1988"	"QiQiHaEr_1988"
"ShenYang_1988"	"TaiYuan_1988"	"TianShui_1988"	"WuLuMuQi_1988"
"XiNing_1988"	"Yi Ning_1988"	"YinChuan_1988"	"YuLin_1988"
"ZhangYe_1988"			

⁽¹⁾Nous utiliserons pour tester cette méthode ainsi que la méthode des nuées dynamiques le projet de programmation co-réalisé notamment par Chavent et Lechevallier dans le cadre du projet européen SODAS. Quelques modifications pratiques y ont été apportées mais aucun changement en profondeur n'a été opéré.

Cluster 2 (Cardinal=39) : [October > 13.200000]

"AnQing_1988"	"BaoDing_1988"	"BeiJing_1988"	"ChangSha_1988"
"ChengDu_1988"	"ChongQing_1988"	"DaLian_1988"	"FuZhou_1988"
"GuangZhou_1988"	"GuiYang_1988"	"HaiKou_1988"	"HangZhou_1988"
"HanZhong_1988"	"JiNan_1988"	"KunMing_1988"	"LiuZhou_1988"
"NanChang_1988"	"NanJing_1988"	"NanNing_1988"	"QingDao_1988"
"QingJiang_1988"	"QuZhou_1988"	"ShangHai_1988"	"ShanTou_1988"
"TengChong_1988"	"TianJin_1988"	"WenZhou_1988"	"WuHan_1988"
"WuZhou_1988"	"XiaMen_1988"	"XiAn_1988"	"XiChang_1988"
"XuZhou_1988"	"YanTai_1988"	"YiChang_1988"	"YongAn_1988"
"ZhanJiang_1988"	"ZhengZhou_1988"	"ZhiJiang_1988"	

Inertie expliquée : 62.223177

PARTITION IN 3 CLUSTERS :

Cluster 1 (Cardinal=21) : [October <= 13.200000]

"BoKeTu_1988"	"ChangChun_1988"	"HaErBin_1988"	"Hailaer_1988"
"HaMi_1988"	"HuHeHaoTe_1988"	"Jiu Quan_1988"	"LanZhou_1988"
"LaSa_1988"	"MuDanJiang_1988"	"NenJiang_1988"	"QiQiHaEr_1988"
"ShenYang_1988"	"TaiYuan_1988"	"TianShui_1988"	"WuLuHuQi_1988"
"XiNing_1988"	"Yi Ning_1988"	"YinChuan_1988"	"YuLin_1988"
"ZhangYe_1988"			

Cluster 2 (Cardinal=24) : [February <= 7.450000]

"AnQing_1988"	"BaoDing_1988"	"BeiJing_1988"	"ChangSha_1988"
"ChengDu_1988"	"DaLian_1988"	"GuiYang_1988"	"HangZhou_1988"
"HanZhong_1988"	"JiNan_1988"	"NanChang_1988"	"NanJing_1988"
"QingDao_1988"	"MuDanJiang_1988"	"QuZhou_1988"	"ShangHai_1988"
"TianJin_1988"	"WuHan_1988"	"XiAn_1988"	"XuZhou_1988"
"YanTai_1988"	"YiChang_1988"	"ZhengZhou_1988"	"ZhiJiang_1988"

Cluster 3 (Cardinal=15) : [February > 7.450000]

"ChongQing_1988"	"FuZhou_1988"	"GuangZhou_1988"	"HaiKou_1988"
"KunMing_1988"	"LiuZhou_1988"	"NanNing_1988"	"ShanTou_1988"
"TengChong_1988"	"WenZhou_1988"	"WuZhou_1988"	"XiaMen_1988"
"XiChang_1988"	"YongAn_1988"	"ZhanJiang_1988"	

Inertie expliquée : 73.998973

PARTITION IN 4 CLUSTERS :

Cluster 1 (Cardinal=7) : [December <= -9.600000]

"BoKeTu_1988"	"ChangChun_1988"	"HaErBin_1988"	"Hailaer_1988"
"MuDanJiang_1988"	"NenJiang_1988"	"QiQiHaEr_1988"	

Cluster 2 (Cardinal=24) : [February <= 7.450000]

"AnQing_1988"	"BaoDing_1988"	"BeiJing_1988"	"ChangSha_1988"
"ChengDu_1988"	"DaLian_1988"	"GuiYang_1988"	"HangZhou_1988"
"HanZhong_1988"	"JiNan_1988"	"NanChang_1988"	"NanJing_1988"
"QingDao_1988"	"QingJiang_1988"	"QuZhou_1988"	"ShangHai_1988"
"TianJin_1988"	"WuHan_1988"	"XiAn_1988"	"XuZhou_1988"
"YanTai_1988"	"YiChang_1988"	"ZhengZhou_1988"	"ZhiJiang_1988"

Cluster 3 (Cardinal=15) : [February > 7.450000]

"ChongQing_1988"	"FuZhou_1988"	"GuangZhou_1988"	"HaiKou_1988"
"KunMing_1988"	"LiuZhou_1988"	"NanNing_1988"	"ShanTou_1988"
"TengChong_1988"	"WenZhou_1988"	"WuZhou_1988"	"XiaMen_1988"
"XiChang_1988"	"YongAn_1988"	"ZhanJiang_1988"	

Cluster 4 (Cardinal=14) : [December > -9.600000]

"HaMi_1988"	"HuHeHaoTe_1988"	"Jiu Quan_1988"	"LanZhou_1988"
"LaSa_1988"	"ShenYang_1988"	"TaiYuan_1988"	"TianShui_1988"
"WuLuHuQi_1988"	"XiNing_1988"	"Yi Ning_1988"	"YinChuan_1988"
"YuLin_1988"	"ZhangYe_1988"		

Inertie expliquée : 82.699077

PARTITION IN 5 CLUSTERS :

```

Cluster 1 (Cardinal=7) : [December <= -9.600000]
"BoKeTu_1988"      "ChangChun_1988"      "HaErBin_1988"      "Hailaer_1988"
"HuDanJiang_1988"  "NenJiang_1988"      "QiQiHaEr_1988"

Cluster 2 (Cardinal=24) : [February <= 7.450000]
"AnQing_1988"      "BaoDing_1988"      "BeiJing_1988"      "ChangSha_1988"
"ChengDu_1988"      "DaLian_1988"      "GuiYang_1988"      "HangZhou_1988"
"HanZhong_1988"      "JiNan_1988"      "NanChang_1988"      "NanJing_1988"
"QingDao_1988"      "QingJiang_1988"    "QuZhou_1988"      "ShangHai_1988"
"TianJin_1988"      "WuHan_1988"      "XiAn_1988"      "XuZhou_1988"
"YanTai_1988"      "YiChang_1988"      "ZhengZhou_1988"    "ZhiJiang_1988"

Cluster 3 (Cardinal=3) : [August <= 25.275000]
"KunMing_1988"      "TengChong_1988"      "XiChang_1988"

Cluster 4 (Cardinal=14) : [December > -9.600000]
"HaMi_1988"      "HuHeHaoTe_1988"      "Jiu Quan_1988"      "LanZhou_1988"
"LaSa_1988"      "ShenYang_1988"      "TaiYuan_1988"      "TianShui_1988"
"WuLuMuQi_1988"  "XiNing_1988"      "Yi Ning_1988"      "YinChuan_1988"
"YuLin_1988"      "ZhangYe_1988"

Cluster 5 (Cardinal=12) : [August > 25.275000]
"ChongQing_1988"  "FuZhou_1988"      "GuangZhou_1988"      "HaiKou_1988"
"LiuZhou_1988"    "NanNing_1988"      "ShanTou_1988"      "WenZhou_1988"
"WuZhou_1988"      "XiaMen_1988"      "YongAn_1988"      "ZhanJiang_1988"

```

Inertie expliquée : 85.758745

THE CLUSTERING TREE :

- the number noted at each node indicates the order of the division
- Ng <-> yes and Nd <-> no

```

+---- Classe 1 (Ng=7)
!
!----3- [December <= -9.600000]
!
! +---- Classe 4 (Nd=14)
!
!----1- [October <= 13.200000]
!
! +---- Classe 2 (Ng=24)
!
!----2- [February <= 7.450000]
!
! +---- Classe 3 (Ng=3)
!
!----4- [August <= 25.275000]
!
! +---- Classe 5 (Nd=12)

```

Nous obtenons successivement les partitions en 2 à 5 classes avec le nombre d'éléments pour chaque classe, la question binaire ayant servi à la division et par conséquent la description symbolique des classes obtenues.

Si nous faisons un graphique de l'évolution de l'inertie expliquée, nous obtenons :

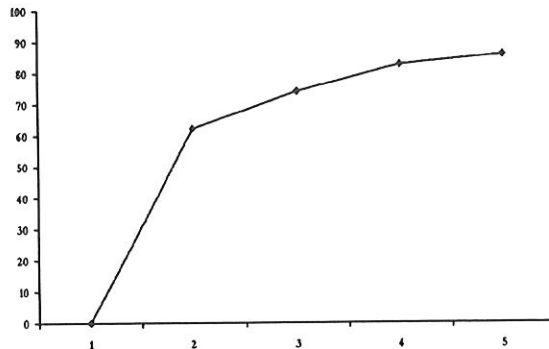


FIGURE 1. Evolution de l'inertie expliquée en fonction du nombre de classes

Le nombre de classes est apparemment 4 puisque l'inertie expliquée supplémentaire engendrée par la création d'une partition à 5 classes est négligeable. Nous nous contenterons de notre intuition, la détermination du nombre de classes constituant une étape assez complexe et très étudiée. De plus, vu le caractère particulier des données, il est difficile de prédire le nombre de classes naturelles (nos connaissances en géographie et en climatologie de la Chine étant limitées).

Nous regroupons ces résultats dans un fichier de sortie créé et formaté à cet effet en vue de la comparaison des méthodes : `chavent.txt` (Annexe C).

10.1.2. Méthode des nuées dynamiques

De la même manière, nous exécutons le programme en choisissant la méthode des nuées dynamiques et le calcul des classifications successives de 2 à 5 classes. Nous obtenons les résultats suivants, extraits du fichier `temp1988.lst` :

PARTITION IN 2 CLUSTERS :

Cluster 1 (Cardinal=39) :

"AnQing_1988"	"BaoDing_1988"	"BeiJing_1988"	"ChangSha_1988"
"ChengDu_1988"	"ChongQing_1988"	"DaLian_1988"	"FuZhou_1988"
"GuangZhou_1988"	"GuiYang_1988"	"HaiKou_1988"	"HangZhou_1988"

"HanZhong_1988"	"JiNan_1988"	"KunMing_1988"	"LiuZhou_1988"
"NanChang_1988"	"NanJing_1988"	"NanMing_1988"	"QingDao_1988"
"QingJiang_1988"	"QuZhou_1988"	"ShangHai_1988"	"ShanTou_1988"
"TengChong_1988"	"TianJin_1988"	"WenZhou_1988"	"WuHan_1988"
"WuZhou_1988"	"XiaMen_1988"	"XiAn_1988"	"XiChang_1988"
"XuZhou_1988"	"YanTai_1988"	"YiChang_1988"	"YongAn_1988"
"ZhanJiang_1988"	"ZhengZhou_1988"	"ZhiJiang_1988"	

Cluster 2 (Cardinal=21) :

"BoKeTu_1988"	"ChangChun_1988"	"HaErBin_1988"	"Hailaer_1988"
"HaMi_1988"	"HuHeHaoTe_1988"	"Jiu Quan_1988"	"LanZhou_1988"
"LaSa_1988"	"MuDanJiang_1988"	"NenJiang_1988"	"QiQiHaEr_1988"
"ShenYang_1988"	"TaiYuan_1988"	"TianShui_1988"	"WuLuMuQi_1988"
"XiNing_1988"	"Yi Ning_1988"	"YinChuan_1988"	"YuLin_1988"
"ZhangYe_1988"			

Inertie expliquée : 61.568430

PARTITION IN 3 CLUSTERS :

Cluster 1 (Cardinal=11) :

"FuZhou_1988"	"GuangZhou_1988"	"HaiKou_1988"	"LiuZhou_1988"
"NanNing_1988"	"ShanTou_1988"	"WuZhou_1988"	"XiaMen_1988"
"XiChang_1988"	"YongAn_1988"	"ZhanJiang_1988"	

Cluster 2 (Cardinal=29) :

"AnQing_1988"	"BaoDing_1988"	"BeiJing_1988"	"ChangSha_1988"
"ChengDu_1988"	"ChongQing_1988"	"DaLian_1988"	"GuiYang_1988"
"HangZhou_1988"	"HanZhong_1988"	"JiNan_1988"	"KunMing_1988"
"NanChang_1988"	"NanJing_1988"	"QingDao_1988"	"QingJiang_1988"
"QuZhou_1988"	"ShangHai_1988"	"TengChong_1988"	"TianJin_1988"
"TianShui_1988"	"WenZhou_1988"	"WuHan_1988"	"XiAn_1988"
"XuZhou_1988"	"YanTai_1988"	"YiChang_1988"	"ZhengZhou_1988"
"ZhiJiang_1988"			

Cluster 3 (Cardinal=20) :

"BoKeTu_1988"	"ChangChun_1988"	"HaErBin_1988"	"Hailaer_1988"
"HaMi_1988"	"HuHeHaoTe_1988"	"Jiu Quan_1988"	"LanZhou_1988"
"LaSa_1988"	"MuDanJiang_1988"	"NenJiang_1988"	"QiQiHaEr_1988"
"ShenYang_1988"	"TaiYuan_1988"	"WuLuMuQi_1988"	"XiNing_1988"
"Yi Ning_1988"	"YinChuan_1988"	"YuLin_1988"	"ZhangYe_1988"

Inertie expliquée : 73.210761

PARTITION IN 4 CLUSTERS :

Cluster 1 (Cardinal=13) :

"HaMi_1988"	"HuHeHaoTe_1988"	"Jiu Quan_1988"	"LanZhou_1988"
"LaSa_1988"	"ShenYang_1988"	"TaiYuan_1988"	"TianShui_1988"
"XiNing_1988"	"Yi Ning_1988"	"YinChuan_1988"	"YuLin_1988"
"ZhangYe_1988"			

Cluster 2 (Cardinal=28) :

"AnQing_1988"	"BaoDing_1988"	"BeiJing_1988"	"ChangSha_1988"
"ChengDu_1988"	"ChongQing_1988"	"DaLian_1988"	"GuiYang_1988"
"HangZhou_1988"	"HanZhong_1988"	"JiNan_1988"	"KunMing_1988"
"NanChang_1988"	"NanJing_1988"	"QingDao_1988"	"QingJiang_1988"
"QuZhou_1988"	"ShangHai_1988"	"TengChong_1988"	"TianJin_1988"
"WenZhou_1988"	"WuHan_1988"	"XiAn_1988"	"XuZhou_1988"
"YanTai_1988"	"YiChang_1988"	"ZhengZhou_1988"	"ZhiJiang_1988"

Cluster 3 (Cardinal=8) :

"BoKeTu_1988"	"ChangChun_1988"	"HaErBin_1988"	"Hailaer_1988"
"MuDanJiang_1988"	"NenJiang_1988"	"QiQiHaEr_1988"	"WuLuMuQi_1988"

Cluster 4 (Cardinal=11) :

"FuZhou_1988"	"GuangZhou_1988"	"HaiKou_1988"	"LiuZhou_1988"
"NanNing_1988"	"ShanTou_1988"	"WuZhou_1988"	"XiaMen_1988"
"XiChang_1988"	"YongAn_1988"	"ZhanJiang_1988"	

Inertie expliquée : 81.935945

PARTITION IN 5 CLUSTERS :

Cluster 1 (Cardinal=7) :

"BoKeTu_1988"	"ChangChun_1988"	"HaErBin_1988"	"Hailaer_1988"
"MuDanJiang_1988"	"NenJiang_1988"	"QiQiHaEr_1988"	

Cluster 2 (Cardinal=16) :

"AnQing_1988"	"ChangSha_1988"	"ChengDu_1988"	"ChongQing_1988"
"GuiYang_1988"	"HangZhou_1988"	"KunMing_1988"	"NanChang_1988"
"NanJing_1988"	"QuZhou_1988"	"ShangHai_1988"	"TengChong_1988"
"WenZhou_1988"	"WuHan_1988"	"YiChang_1988"	"ZhiJiang_1988"

Cluster 3 (Cardinal=13) :

"HaMi_1988"	"HuHeHaoTe_1988"	"Jiu Quan_1988"	"LanZhou_1988"
"LaSa_1988"	"ShenYang_1988"	"TaiYuan_1988"	"WuLuMuQi_1988"
"XiNing_1988"	"Yi Ning_1988"	"YinChuan_1988"	"YuLin_1988"
"ZhangYe_1988"			

Cluster 4 (Cardinal=13) :

"BaoDing_1988"	"BeiJing_1988"	"DaLian_1988"	"HanZhong_1988"
"JiNan_1988"	"QingDao_1988"	"QingJiang_1988"	"TianJin_1988"
"TianShui_1988"	"XiAn_1988"	"XuZhou_1988"	"YanTai_1988"
"ZhengZhou_1988"			

Cluster 5 (Cardinal=11) :

"FuZhou_1988"	"GuangZhou_1988"	"HaiKou_1988"	"LiuZhou_1988"
"NanNing_1988"	"ShanTou_1988"	"WuZhou_1988"	"XiaMen_1988"
"XiChang_1988"	"YongAn_1988"	"ZhanJiang_1988"	

Inertie expliquée : 86.006085

Si nous faisons un graphique de l'évolution de l'inertie expliquée, nous obtenons :

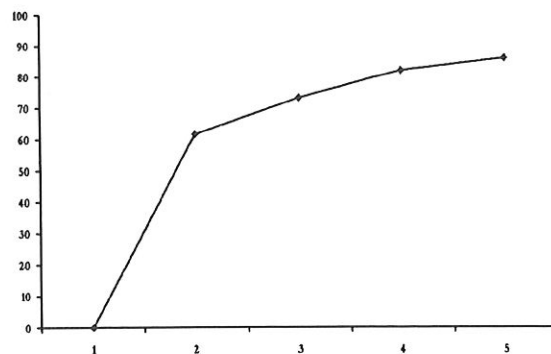


FIGURE 2. Evolution de l'inertie expliquée en fonction du nombre de classes

Le nombre de classes est apparemment 4 pour les mêmes raisons que vues précédemment.

Nous regroupons également ces résultats dans un fichier de sortie créé et formaté à cet effet en vue de la comparaison des méthodes : `scluster.txt` (Annexe C).

10.1.3. Comparaison des méthodes

D'abord, on peut montrer que l'efficacité des deux méthodes est à peu de choses presque identique puisque l'évolution de l'inertie expliquée en fonction du nombre de classes est semblable. Ces inerties sont reprises dans la Table 1.

	Méthode de Chavent	Nuées dynamiques
2 classes	62.22	61.57
3 classes	74.00	73.21
4 classes	82.70	81.94
5 classes	85.76	86.00

Table 1: Inerties expliquées selon les méthodes

Ensuite, on applique le programme de comparaison (`comparaison.cpp`) aux fichiers de données créés précédemment (`chavent.txt` et `scluster.txt`) et on crée le fichier de résultats : `resultat.txt`.

Matrice de contingence pour 2 classes :

```
0      39
21     0
```

Valeur de chi2 pour 2 classes : 70.800000

Valeur de max pour 2 classes : 60 / 60 => Pourcentage de bien classes : 100

Matrice de contingence pour 3 classes :

```
0      0      11
1     24       4
20     0       0
```

Valeur de chi2 pour 3 classes : 107.100000

Valeur de max pour 3 classes : 55 / 60 => Pourcentage de bien classes : 91

Matrice de contingence pour 4 classes :

```
0      0      0      13
0     24      4       0
```

7	0	0	1
0	0	11	0

Valeur de chi2 pour 4 classes : 188.533333

Valeur de max pour 4 classes : 55 / 60 => Pourcentage de bien classes : 91

Matrice de contingence pour 5 classes :

7	0	0	0	0
0	12	2	0	2
0	0	0	13	0
0	12	0	1	0
0	0	1	0	10

Valeur de chi2 pour 5 classes : 196.666667

Valeur de max pour 5 classes : 42 / 60 => Pourcentage de bien classes : 70

Les valeurs du test du χ^2 d'indépendance ne sont pas vraiment intéressantes car, comme on pouvait s'y attendre, elles sont très élevées (en fonction du degré de liberté correspondant : 1, 4, 9, 16) confirmant la non-indépendance des éléments du tableau de contingence. La maximisation "ligne-colonne" est, par contre, bien plus intéressante. On voit bien que les partitions en 2 classes issues des deux méthodes sont identiques et que pour un nombre de classes plus élevé elles restent fort semblables (plus de 90% jusqu'à 4 classes). Au delà du nombre de classes naturelles que nous avons déduit de notre intuition, les partitions commencent à se démarquer l'une de l'autre. Le pourcentage de "bien classés" appartenant à des classes assez semblables obtenues par les méthodes diminue fortement et n'est plus que de 70%.

CHAPITRE 11

APPLICATION : DONNÉES ARTIFICIELLES

11.1. Création de données artificielles

Pour ne plus avoir les inconvénients d'interprétation des données réelles, nous avons créé une petite application permettant de générer des données artificielles. La création des données artificielles est assez simple. On va en fait créer dans l'espace des hypercubes qui serviront d'enveloppes à nos objets symboliques intervalles.

On commence par demander à l'utilisateur :

- la dimension de l'espace de représentation, c'est-à-dire, le nombre de variables caractérisant chaque individu (p).
- le nombre de classes que l'utilisateur veut artificiellement créer (m).
- le nombre d'intervalles créés pour chaque classe (n) d'où $n \times m$ individus dans le jeu de données.
- une valeur qui nous permettra de calculer le centre de gravité des hypercubes (g).
- la longueur maximale des intervalles (l).

Pour chaque individus, nous devons calculer p variables intervalles correspondant à chacune des dimensions. On regarde simplement les coordonnées

du centre de gravité de l'hypercube. Soit g_j la coordonnée pour la j^e dimension ($j = 1, \dots, p$), afin de donner un caractère aléatoire à notre intervalle, nous choisirons le milieu de notre intervalle en créant une loi uniforme $milieu_j \sim U[g_j - \frac{l}{2}, g_j + \frac{l}{2}]$. Cela revient à choisir la valeur de $milieu_j$ dans l'intervalle $[g_j - \frac{l}{2}, g_j + \frac{l}{2}]$. Pour donner, un caractère aléatoire à la longueur de l'intervalle, nous choisirons la longueur de notre intervalle en créant une loi uniforme $long_j \sim U[0, l]$. Cela revient à choisir la valeur de $long_j$ dans l'intervalle $[0, l]$. L'intervalle ainsi construit est donc inclus dans l'intervalle maximal : $[g_j - \frac{l}{2} - \frac{l}{2}, g_j + \frac{l}{2} + \frac{l}{2}] = [g_j - l, g_j + l]$. C'est pourquoi lorsque nous créons une classe, les individus se trouvent dans un hypercube à p dimensions dont le centre de gravité est g et dont les arêtes sont de longueur $2 * l$. Les classes se différencieront simplement par un centre de gravité de coordonnées différentes.

Ainsi, dans le premier jeu de données créés par notre programme : nous choisissons comme valeur d'initialisation : $p = 5$, $m = 4$, une valeur de 1 comme milieu moyen en valeur absolue, et $l = 1$. Notre programme crée alors 4 vecteurs à 5 dimensions pour les 4 centres de gravité des classes $((1, 1, 1, 1, 1), (1, 1, 1, 1, -1), (1, 1, 1, -1, 1)$ et $(1, 1, 1, -1, -1))$. Ces 4 centres de gravité définissent 4 hypercubes disjoints dans \mathbb{R}^5 . Remarquons que certains de ces hypercubes sont contigus vus les données d'initialisation. Pour notre second jeu de données que nous utiliserons lors de la comparaison, on choisira de remplacer la valeur du milieu moyen par 2, cela entraîne la séparation spatiale des hypercubes. Les résultats pour le premier jeu de données sont données dans le fichier `liste.txt`.

Pour en faire un fichier SODAS, nous devons transformer ce fichier texte en une base de données Access. Nous ouvrons donc une nouvelle base Access que nous appelons `liste.mdb` et dans le menu Fichier, nous choisissons le sous-menu Données externes puis Importer.... On choisit alors d'importer le fichier `liste.txt` et on effectuera les opérations montrées ci-dessous :

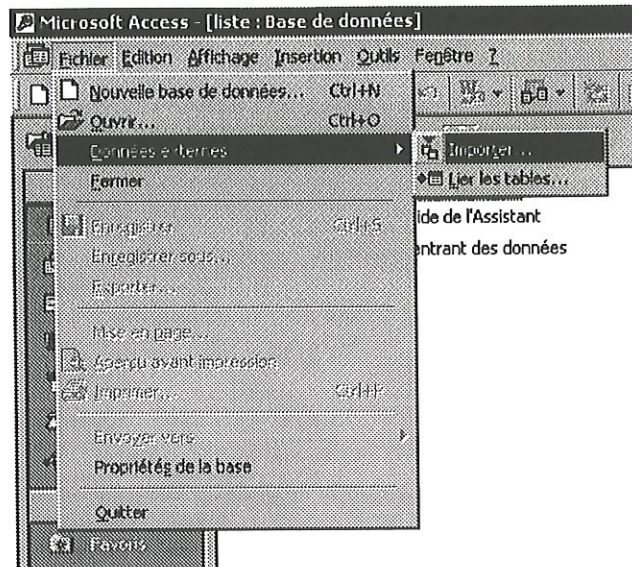


FIGURE 1. Menu Importer...

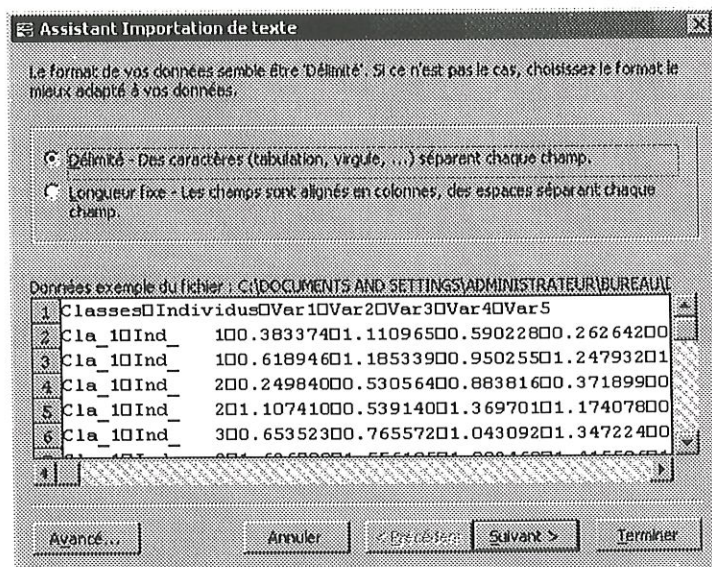


FIGURE 2. Premier menu d'importation

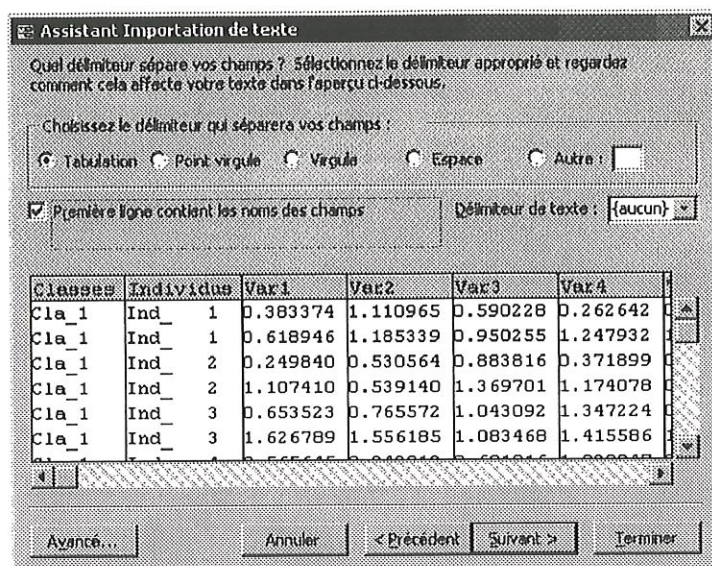


FIGURE 3. Second menu d'importation

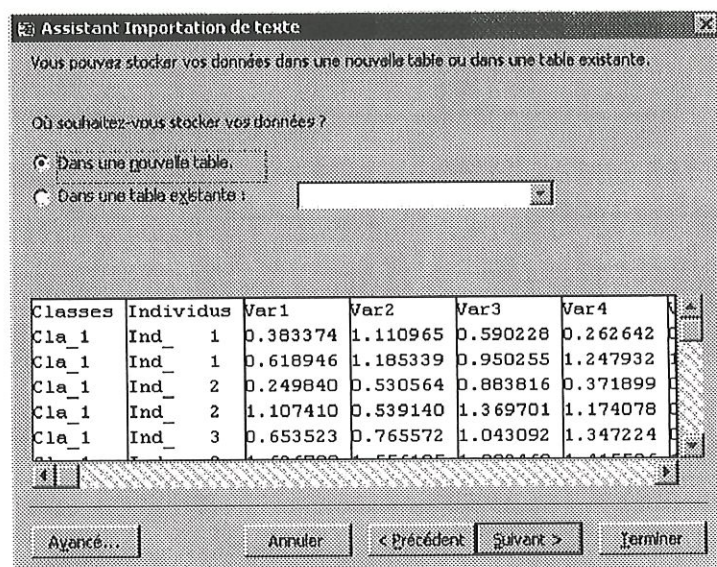


FIGURE 4. Troisième menu d'importation

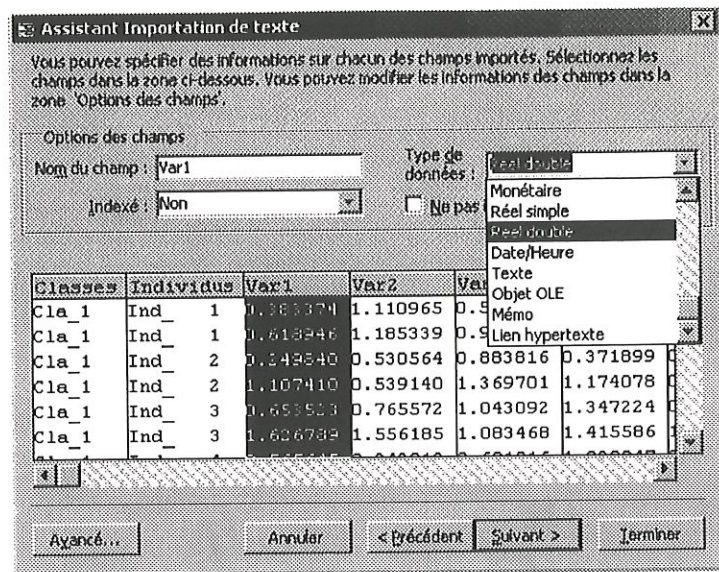


FIGURE 5. Quatrième menu d'importation (Texte → Réel double pour les variables)

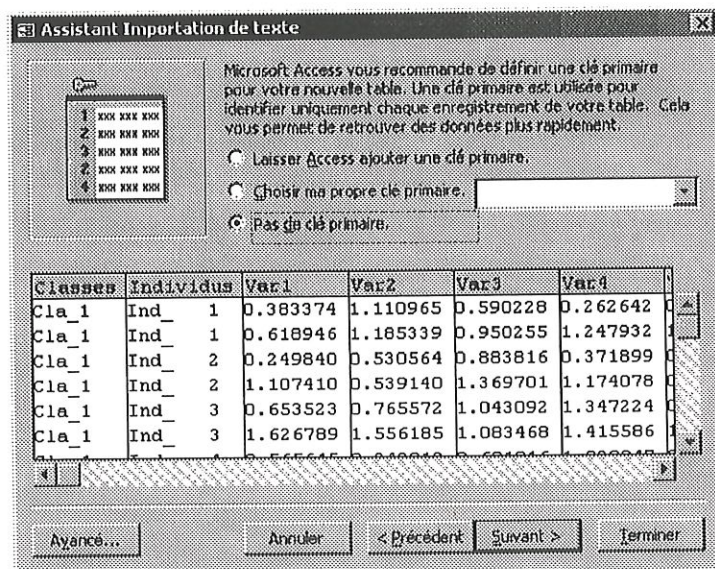


FIGURE 6. Cinquième menu d'importation

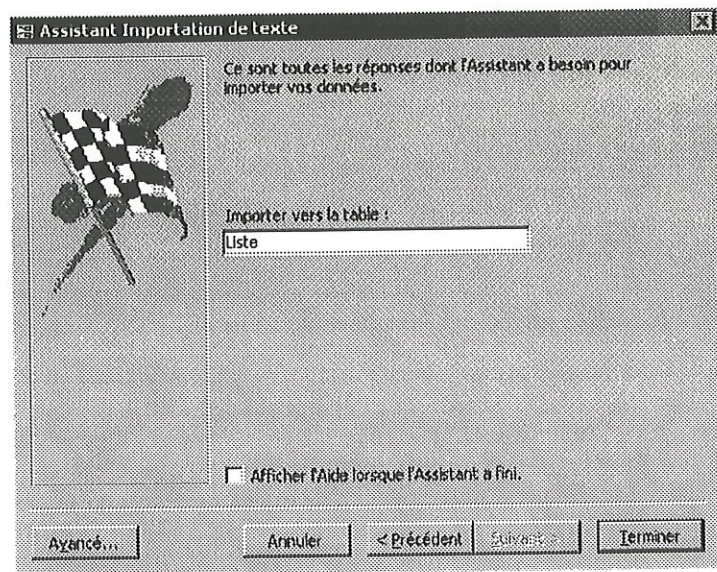


FIGURE 7. Dernier menu d'importation

Cette importation terminée, il faut maintenant transformer cette base de donnée Access en fichier SODAS. Pour cela, on utilisera le programme DB2SO [2] en choisissant le menu File puis le sous-menu New query :

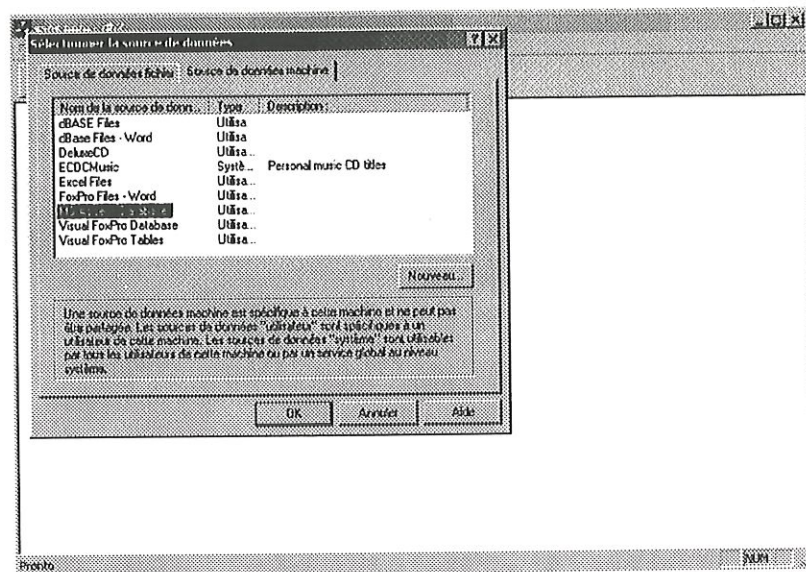


FIGURE 8. Choix d'une base MS Access

On choisit le fichier `liste.mdb` puis :

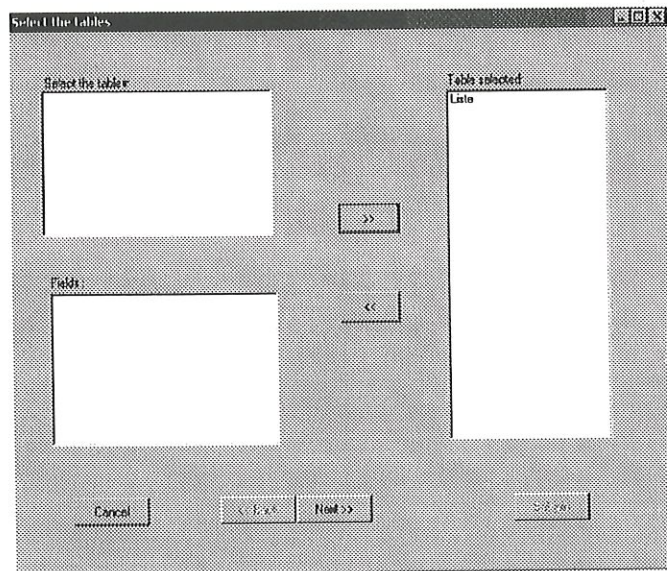


FIGURE 9. Choix de la base de données Liste

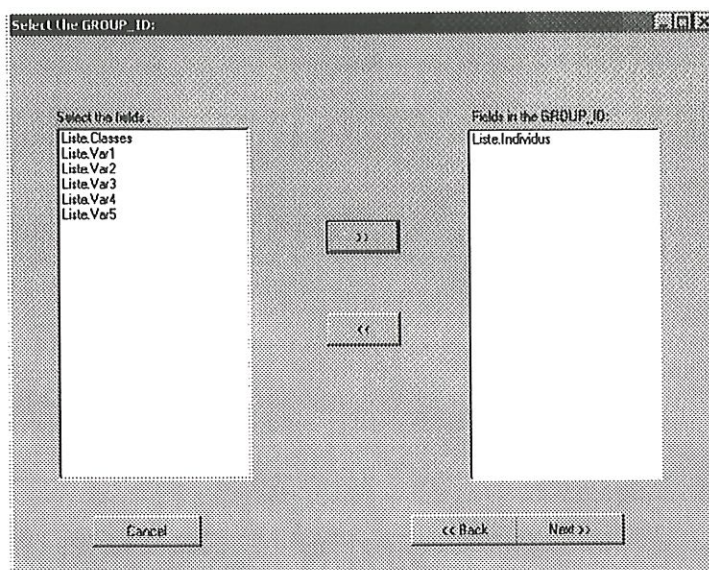


FIGURE 10. Sélection de l'identification des individus

Les individus sont identifiés selon le champs Liste.Individus

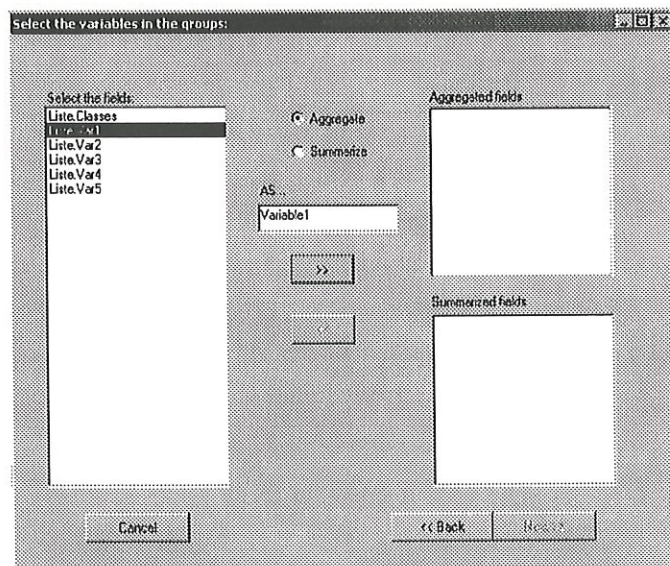


FIGURE 11. Sélection des variables

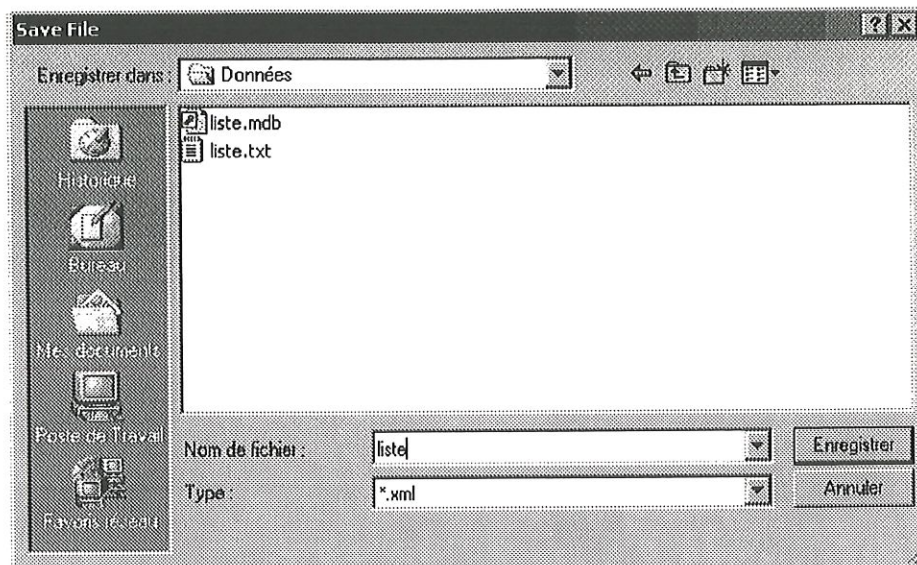


FIGURE 12. Sauvegarde de cette Query sous le nom liste.xml

On ouvre ensuite cette Query (liste.xml) via le menu File et le sous-menu Open query Une fois avoir cliqué l'option Extract SO..., un fichier SODAS est créé et nous le sauvons sous le nom de `artificielles.sds`.

11.2. Jeu de données "artificielles.sds"

11.2.1. La méthode divisive de Chavent

Comme pour le jeu de données vu précédemment, nous appliquons la méthode DIV de Chavent en faisant les choix suivants : une division jusqu'en 5 classes et une non-normalisation de la distance puisque les variables sont de même type. Nous obtenons les résultats suivants, extraits du fichier `artificielles.lst` :

PARTITION IN 2 CLUSTERS :

```
Cluster 1 (Cardinal=200) : [Variable5 <= -0.003006]
"Ind_ 101"    "Ind_ 102"    "Ind_ 103"    "Ind_ 104"
"Ind_ 105"    "Ind_ 106"    "Ind_ 107"    "Ind_ 108"
"Ind_ 109"    "Ind_ 110"    "Ind_ 111"    "Ind_ 112"
"Ind_ 113"    "Ind_ 114"    "Ind_ 115"    "Ind_ 116"
"Ind_ 117"    "Ind_ 118"    "Ind_ 119"    "Ind_ 120"
"Ind_ 121"    "Ind_ 122"    "Ind_ 123"    "Ind_ 124"
"Ind_ 125"    "Ind_ 126"    "Ind_ 127"    "Ind_ 128"
"Ind_ 129"    "Ind_ 130"    "Ind_ 131"    "Ind_ 132"
```

"Ind_ 133"	"Ind_ 134"	"Ind_ 135"	"Ind_ 136"
"Ind_ 137"	"Ind_ 138"	"Ind_ 139"	"Ind_ 140"
"Ind_ 141"	"Ind_ 142"	"Ind_ 143"	"Ind_ 144"
"Ind_ 145"	"Ind_ 146"	"Ind_ 147"	"Ind_ 148"
"Ind_ 149"	"Ind_ 150"	"Ind_ 151"	"Ind_ 152"
"Ind_ 153"	"Ind_ 154"	"Ind_ 155"	"Ind_ 156"
"Ind_ 157"	"Ind_ 158"	"Ind_ 159"	"Ind_ 160"
"Ind_ 161"	"Ind_ 162"	"Ind_ 163"	"Ind_ 164"
"Ind_ 165"	"Ind_ 166"	"Ind_ 167"	"Ind_ 168"
"Ind_ 169"	"Ind_ 170"	"Ind_ 171"	"Ind_ 172"
"Ind_ 173"	"Ind_ 174"	"Ind_ 175"	"Ind_ 176"
"Ind_ 177"	"Ind_ 178"	"Ind_ 179"	"Ind_ 180"
"Ind_ 181"	"Ind_ 182"	"Ind_ 183"	"Ind_ 184"
"Ind_ 185"	"Ind_ 186"	"Ind_ 187"	"Ind_ 188"
"Ind_ 189"	"Ind_ 190"	"Ind_ 191"	"Ind_ 192"
"Ind_ 193"	"Ind_ 194"	"Ind_ 195"	"Ind_ 196"
"Ind_ 197"	"Ind_ 198"	"Ind_ 199"	"Ind_ 200"
"Ind_ 301"	"Ind_ 302"	"Ind_ 303"	"Ind_ 304"
"Ind_ 305"	"Ind_ 306"	"Ind_ 307"	"Ind_ 308"
"Ind_ 309"	"Ind_ 310"	"Ind_ 311"	"Ind_ 312"
"Ind_ 313"	"Ind_ 314"	"Ind_ 315"	"Ind_ 316"
"Ind_ 317"	"Ind_ 318"	"Ind_ 319"	"Ind_ 320"
"Ind_ 321"	"Ind_ 322"	"Ind_ 323"	"Ind_ 324"
"Ind_ 325"	"Ind_ 326"	"Ind_ 327"	"Ind_ 328"
"Ind_ 329"	"Ind_ 330"	"Ind_ 331"	"Ind_ 332"
"Ind_ 333"	"Ind_ 334"	"Ind_ 335"	"Ind_ 336"
"Ind_ 337"	"Ind_ 338"	"Ind_ 339"	"Ind_ 340"
"Ind_ 341"	"Ind_ 342"	"Ind_ 343"	"Ind_ 344"
"Ind_ 345"	"Ind_ 346"	"Ind_ 347"	"Ind_ 348"
"Ind_ 349"	"Ind_ 350"	"Ind_ 351"	"Ind_ 352"
"Ind_ 353"	"Ind_ 354"	"Ind_ 355"	"Ind_ 356"
"Ind_ 357"	"Ind_ 358"	"Ind_ 359"	"Ind_ 360"
"Ind_ 361"	"Ind_ 362"	"Ind_ 363"	"Ind_ 364"
"Ind_ 365"	"Ind_ 366"	"Ind_ 367"	"Ind_ 368"
"Ind_ 369"	"Ind_ 370"	"Ind_ 371"	"Ind_ 372"
"Ind_ 373"	"Ind_ 374"	"Ind_ 375"	"Ind_ 376"
"Ind_ 377"	"Ind_ 378"	"Ind_ 379"	"Ind_ 380"
"Ind_ 381"	"Ind_ 382"	"Ind_ 383"	"Ind_ 384"
"Ind_ 385"	"Ind_ 386"	"Ind_ 387"	"Ind_ 388"
"Ind_ 389"	"Ind_ 390"	"Ind_ 391"	"Ind_ 392"
"Ind_ 393"	"Ind_ 394"	"Ind_ 395"	"Ind_ 396"
"Ind_ 397"	"Ind_ 398"	"Ind_ 399"	"Ind_ 400"

Cluster 2 (Cardinal=200) : [Variable5 > -0.003006]

"Ind_ 1"	"Ind_ 2"	"Ind_ 3"	"Ind_ 4"
"Ind_ 5"	"Ind_ 6"	"Ind_ 7"	"Ind_ 8"
"Ind_ 9"	"Ind_ 10"	"Ind_ 11"	"Ind_ 12"
"Ind_ 13"	"Ind_ 14"	"Ind_ 15"	"Ind_ 16"
"Ind_ 17"	"Ind_ 18"	"Ind_ 19"	"Ind_ 20"
"Ind_ 21"	"Ind_ 22"	"Ind_ 23"	"Ind_ 24"
"Ind_ 25"	"Ind_ 26"	"Ind_ 27"	"Ind_ 28"
"Ind_ 29"	"Ind_ 30"	"Ind_ 31"	"Ind_ 32"
"Ind_ 33"	"Ind_ 34"	"Ind_ 35"	"Ind_ 36"
"Ind_ 37"	"Ind_ 38"	"Ind_ 39"	"Ind_ 40"
"Ind_ 41"	"Ind_ 42"	"Ind_ 43"	"Ind_ 44"
"Ind_ 45"	"Ind_ 46"	"Ind_ 47"	"Ind_ 48"
"Ind_ 49"	"Ind_ 50"	"Ind_ 51"	"Ind_ 52"
"Ind_ 53"	"Ind_ 54"	"Ind_ 55"	"Ind_ 56"
"Ind_ 57"	"Ind_ 58"	"Ind_ 59"	"Ind_ 60"
"Ind_ 61"	"Ind_ 62"	"Ind_ 63"	"Ind_ 64"
"Ind_ 65"	"Ind_ 66"	"Ind_ 67"	"Ind_ 68"
"Ind_ 69"	"Ind_ 70"	"Ind_ 71"	"Ind_ 72"
"Ind_ 73"	"Ind_ 74"	"Ind_ 75"	"Ind_ 76"
"Ind_ 77"	"Ind_ 78"	"Ind_ 79"	"Ind_ 80"
"Ind_ 81"	"Ind_ 82"	"Ind_ 83"	"Ind_ 84"
"Ind_ 85"	"Ind_ 86"	"Ind_ 87"	"Ind_ 88"
"Ind_ 89"	"Ind_ 90"	"Ind_ 91"	"Ind_ 92"

"Ind_ 93"	"Ind_ 94"	"Ind_ 95"	"Ind_ 96"
"Ind_ 97"	"Ind_ 98"	"Ind_ 99"	"Ind_ 100"
"Ind_ 201"	"Ind_ 202"	"Ind_ 203"	"Ind_ 204"
"Ind_ 205"	"Ind_ 206"	"Ind_ 207"	"Ind_ 208"
"Ind_ 209"	"Ind_ 210"	"Ind_ 211"	"Ind_ 212"
"Ind_ 213"	"Ind_ 214"	"Ind_ 215"	"Ind_ 216"
"Ind_ 217"	"Ind_ 218"	"Ind_ 219"	"Ind_ 220"
"Ind_ 221"	"Ind_ 222"	"Ind_ 223"	"Ind_ 224"
"Ind_ 225"	"Ind_ 226"	"Ind_ 227"	"Ind_ 228"
"Ind_ 229"	"Ind_ 230"	"Ind_ 231"	"Ind_ 232"
"Ind_ 233"	"Ind_ 234"	"Ind_ 235"	"Ind_ 236"
"Ind_ 237"	"Ind_ 238"	"Ind_ 239"	"Ind_ 240"
"Ind_ 241"	"Ind_ 242"	"Ind_ 243"	"Ind_ 244"
"Ind_ 245"	"Ind_ 246"	"Ind_ 247"	"Ind_ 248"
"Ind_ 249"	"Ind_ 250"	"Ind_ 251"	"Ind_ 252"
"Ind_ 253"	"Ind_ 254"	"Ind_ 255"	"Ind_ 256"
"Ind_ 257"	"Ind_ 258"	"Ind_ 259"	"Ind_ 260"
"Ind_ 261"	"Ind_ 262"	"Ind_ 263"	"Ind_ 264"
"Ind_ 265"	"Ind_ 266"	"Ind_ 267"	"Ind_ 268"
"Ind_ 269"	"Ind_ 270"	"Ind_ 271"	"Ind_ 272"
"Ind_ 273"	"Ind_ 274"	"Ind_ 275"	"Ind_ 276"
"Ind_ 277"	"Ind_ 278"	"Ind_ 279"	"Ind_ 280"
"Ind_ 281"	"Ind_ 282"	"Ind_ 283"	"Ind_ 284"
"Ind_ 285"	"Ind_ 286"	"Ind_ 287"	"Ind_ 288"
"Ind_ 289"	"Ind_ 290"	"Ind_ 291"	"Ind_ 292"
"Ind_ 293"	"Ind_ 294"	"Ind_ 295"	"Ind_ 296"
"Ind_ 297"	"Ind_ 298"	"Ind_ 299"	"Ind_ 300"

Inertie expliquée : 37.791146

PARTITION IN 3 CLUSTERS :

```

Cluster 1 (Cardinal=200) : [Variable5 <= -0.003006]
"Ind_ 101"    "Ind_ 102"    "Ind_ 103"    "Ind_ 104"
"Ind_ 105"    "Ind_ 106"    "Ind_ 107"    "Ind_ 108"
"Ind_ 109"    "Ind_ 110"    "Ind_ 111"    "Ind_ 112"
"Ind_ 113"    "Ind_ 114"    "Ind_ 115"    "Ind_ 116"
"Ind_ 117"    "Ind_ 118"    "Ind_ 119"    "Ind_ 120"
"Ind_ 121"    "Ind_ 122"    "Ind_ 123"    "Ind_ 124"
"Ind_ 125"    "Ind_ 126"    "Ind_ 127"    "Ind_ 128"
"Ind_ 129"    "Ind_ 130"    "Ind_ 131"    "Ind_ 132"
"Ind_ 133"    "Ind_ 134"    "Ind_ 135"    "Ind_ 136"
"Ind_ 137"    "Ind_ 138"    "Ind_ 139"    "Ind_ 140"
"Ind_ 141"    "Ind_ 142"    "Ind_ 143"    "Ind_ 144"
"Ind_ 145"    "Ind_ 146"    "Ind_ 147"    "Ind_ 148"
"Ind_ 149"    "Ind_ 150"    "Ind_ 151"    "Ind_ 152"
"Ind_ 153"    "Ind_ 154"    "Ind_ 155"    "Ind_ 156"
"Ind_ 157"    "Ind_ 158"    "Ind_ 159"    "Ind_ 160"
"Ind_ 161"    "Ind_ 162"    "Ind_ 163"    "Ind_ 164"
"Ind_ 165"    "Ind_ 166"    "Ind_ 167"    "Ind_ 168"
"Ind_ 169"    "Ind_ 170"    "Ind_ 171"    "Ind_ 172"
"Ind_ 173"    "Ind_ 174"    "Ind_ 175"    "Ind_ 176"
"Ind_ 177"    "Ind_ 178"    "Ind_ 179"    "Ind_ 180"
"Ind_ 181"    "Ind_ 182"    "Ind_ 183"    "Ind_ 184"
"Ind_ 185"    "Ind_ 186"    "Ind_ 187"    "Ind_ 188"
"Ind_ 189"    "Ind_ 190"    "Ind_ 191"    "Ind_ 192"
"Ind_ 193"    "Ind_ 194"    "Ind_ 195"    "Ind_ 196"
"Ind_ 197"    "Ind_ 198"    "Ind_ 199"    "Ind_ 200"
"Ind_ 301"    "Ind_ 302"    "Ind_ 303"    "Ind_ 304"
"Ind_ 305"    "Ind_ 306"    "Ind_ 307"    "Ind_ 308"
"Ind_ 309"    "Ind_ 310"    "Ind_ 311"    "Ind_ 312"
"Ind_ 313"    "Ind_ 314"    "Ind_ 315"    "Ind_ 316"
"Ind_ 317"    "Ind_ 318"    "Ind_ 319"    "Ind_ 320"
"Ind_ 321"    "Ind_ 322"    "Ind_ 323"    "Ind_ 324"
"Ind_ 325"    "Ind_ 326"    "Ind_ 327"    "Ind_ 328"
"Ind_ 329"    "Ind_ 330"    "Ind_ 331"    "Ind_ 332"

```

"Ind_ 333"	"Ind_ 334"	"Ind_ 335"	"Ind_ 336"
"Ind_ 337"	"Ind_ 338"	"Ind_ 339"	"Ind_ 340"
"Ind_ 341"	"Ind_ 342"	"Ind_ 343"	"Ind_ 344"
"Ind_ 345"	"Ind_ 346"	"Ind_ 347"	"Ind_ 348"
"Ind_ 349"	"Ind_ 350"	"Ind_ 351"	"Ind_ 352"
"Ind_ 353"	"Ind_ 354"	"Ind_ 355"	"Ind_ 356"
"Ind_ 357"	"Ind_ 358"	"Ind_ 359"	"Ind_ 360"
"Ind_ 361"	"Ind_ 362"	"Ind_ 363"	"Ind_ 364"
"Ind_ 365"	"Ind_ 366"	"Ind_ 367"	"Ind_ 368"
"Ind_ 369"	"Ind_ 370"	"Ind_ 371"	"Ind_ 372"
"Ind_ 373"	"Ind_ 374"	"Ind_ 375"	"Ind_ 376"
"Ind_ 377"	"Ind_ 378"	"Ind_ 379"	"Ind_ 380"
"Ind_ 381"	"Ind_ 382"	"Ind_ 383"	"Ind_ 384"
"Ind_ 385"	"Ind_ 386"	"Ind_ 387"	"Ind_ 388"
"Ind_ 389"	"Ind_ 390"	"Ind_ 391"	"Ind_ 392"
"Ind_ 393"	"Ind_ 394"	"Ind_ 395"	"Ind_ 396"
"Ind_ 397"	"Ind_ 398"	"Ind_ 399"	"Ind_ 400"

Cluster 2 (Cardinal=100) : [Variable4 <= -0.000977]

"Ind_ 201"	"Ind_ 202"	"Ind_ 203"	"Ind_ 204"
"Ind_ 205"	"Ind_ 206"	"Ind_ 207"	"Ind_ 208"
"Ind_ 209"	"Ind_ 210"	"Ind_ 211"	"Ind_ 212"
"Ind_ 213"	"Ind_ 214"	"Ind_ 215"	"Ind_ 216"
"Ind_ 217"	"Ind_ 218"	"Ind_ 219"	"Ind_ 220"
"Ind_ 221"	"Ind_ 222"	"Ind_ 223"	"Ind_ 224"
"Ind_ 225"	"Ind_ 226"	"Ind_ 227"	"Ind_ 228"
"Ind_ 229"	"Ind_ 230"	"Ind_ 231"	"Ind_ 232"
"Ind_ 233"	"Ind_ 234"	"Ind_ 235"	"Ind_ 236"
"Ind_ 237"	"Ind_ 238"	"Ind_ 239"	"Ind_ 240"
"Ind_ 241"	"Ind_ 242"	"Ind_ 243"	"Ind_ 244"
"Ind_ 245"	"Ind_ 246"	"Ind_ 247"	"Ind_ 248"
"Ind_ 249"	"Ind_ 250"	"Ind_ 251"	"Ind_ 252"
"Ind_ 253"	"Ind_ 254"	"Ind_ 255"	"Ind_ 256"
"Ind_ 257"	"Ind_ 258"	"Ind_ 259"	"Ind_ 260"
"Ind_ 261"	"Ind_ 262"	"Ind_ 263"	"Ind_ 264"
"Ind_ 265"	"Ind_ 266"	"Ind_ 267"	"Ind_ 268"
"Ind_ 269"	"Ind_ 270"	"Ind_ 271"	"Ind_ 272"
"Ind_ 273"	"Ind_ 274"	"Ind_ 275"	"Ind_ 276"
"Ind_ 277"	"Ind_ 278"	"Ind_ 279"	"Ind_ 280"
"Ind_ 281"	"Ind_ 282"	"Ind_ 283"	"Ind_ 284"
"Ind_ 285"	"Ind_ 286"	"Ind_ 287"	"Ind_ 288"
"Ind_ 289"	"Ind_ 290"	"Ind_ 291"	"Ind_ 292"
"Ind_ 293"	"Ind_ 294"	"Ind_ 295"	"Ind_ 296"
"Ind_ 297"	"Ind_ 298"	"Ind_ 299"	"Ind_ 300"

Cluster 3 (Cardinal=100) : [Variable4 > -0.000977]

"Ind_ 1"	"Ind_ 2"	"Ind_ 3"	"Ind_ 4"
"Ind_ 5"	"Ind_ 6"	"Ind_ 7"	"Ind_ 8"
"Ind_ 9"	"Ind_ 10"	"Ind_ 11"	"Ind_ 12"
"Ind_ 13"	"Ind_ 14"	"Ind_ 15"	"Ind_ 16"
"Ind_ 17"	"Ind_ 18"	"Ind_ 19"	"Ind_ 20"
"Ind_ 21"	"Ind_ 22"	"Ind_ 23"	"Ind_ 24"
"Ind_ 25"	"Ind_ 26"	"Ind_ 27"	"Ind_ 28"
"Ind_ 29"	"Ind_ 30"	"Ind_ 31"	"Ind_ 32"
"Ind_ 33"	"Ind_ 34"	"Ind_ 35"	"Ind_ 36"
"Ind_ 37"	"Ind_ 38"	"Ind_ 39"	"Ind_ 40"
"Ind_ 41"	"Ind_ 42"	"Ind_ 43"	"Ind_ 44"
"Ind_ 45"	"Ind_ 46"	"Ind_ 47"	"Ind_ 48"
"Ind_ 49"	"Ind_ 50"	"Ind_ 51"	"Ind_ 52"
"Ind_ 53"	"Ind_ 54"	"Ind_ 55"	"Ind_ 56"
"Ind_ 57"	"Ind_ 58"	"Ind_ 59"	"Ind_ 60"
"Ind_ 61"	"Ind_ 62"	"Ind_ 63"	"Ind_ 64"
"Ind_ 65"	"Ind_ 66"	"Ind_ 67"	"Ind_ 68"
"Ind_ 69"	"Ind_ 70"	"Ind_ 71"	"Ind_ 72"
"Ind_ 73"	"Ind_ 74"	"Ind_ 75"	"Ind_ 76"
"Ind_ 77"	"Ind_ 78"	"Ind_ 79"	"Ind_ 80"
"Ind_ 81"	"Ind_ 82"	"Ind_ 83"	"Ind_ 84"

"Ind_ 85"	"Ind_ 86"	"Ind_ 87"	"Ind_ 88"
"Ind_ 89"	"Ind_ 90"	"Ind_ 91"	"Ind_ 92"
"Ind_ 93"	"Ind_ 94"	"Ind_ 95"	"Ind_ 96"
"Ind_ 97"	"Ind_ 98"	"Ind_ 99"	"Ind_ 100"

Inertie expliquée : 56.278118

PARTITION IN 4 CLUSTERS :

```

-----
Cluster 1 (Cardinal=100) : [Variable4 <= -0.000824]
"Ind_ 301"  "Ind_ 302"  "Ind_ 303"  "Ind_ 304"
"Ind_ 305"  "Ind_ 306"  "Ind_ 307"  "Ind_ 308"
"Ind_ 309"  "Ind_ 310"  "Ind_ 311"  "Ind_ 312"
"Ind_ 313"  "Ind_ 314"  "Ind_ 315"  "Ind_ 316"
"Ind_ 317"  "Ind_ 318"  "Ind_ 319"  "Ind_ 320"
"Ind_ 321"  "Ind_ 322"  "Ind_ 323"  "Ind_ 324"
"Ind_ 325"  "Ind_ 326"  "Ind_ 327"  "Ind_ 328"
"Ind_ 329"  "Ind_ 330"  "Ind_ 331"  "Ind_ 332"
"Ind_ 333"  "Ind_ 334"  "Ind_ 335"  "Ind_ 336"
"Ind_ 337"  "Ind_ 338"  "Ind_ 339"  "Ind_ 340"
"Ind_ 341"  "Ind_ 342"  "Ind_ 343"  "Ind_ 344"
"Ind_ 345"  "Ind_ 346"  "Ind_ 347"  "Ind_ 348"
"Ind_ 349"  "Ind_ 350"  "Ind_ 351"  "Ind_ 352"
"Ind_ 353"  "Ind_ 354"  "Ind_ 355"  "Ind_ 356"
"Ind_ 357"  "Ind_ 358"  "Ind_ 359"  "Ind_ 360"
"Ind_ 361"  "Ind_ 362"  "Ind_ 363"  "Ind_ 364"
"Ind_ 365"  "Ind_ 366"  "Ind_ 367"  "Ind_ 368"
"Ind_ 369"  "Ind_ 370"  "Ind_ 371"  "Ind_ 372"
"Ind_ 373"  "Ind_ 374"  "Ind_ 375"  "Ind_ 376"
"Ind_ 377"  "Ind_ 378"  "Ind_ 379"  "Ind_ 380"
"Ind_ 381"  "Ind_ 382"  "Ind_ 383"  "Ind_ 384"
"Ind_ 385"  "Ind_ 386"  "Ind_ 387"  "Ind_ 388"
"Ind_ 389"  "Ind_ 390"  "Ind_ 391"  "Ind_ 392"
"Ind_ 393"  "Ind_ 394"  "Ind_ 395"  "Ind_ 396"
"Ind_ 397"  "Ind_ 398"  "Ind_ 399"  "Ind_ 400"

Cluster 2 (Cardinal=100) : [Variable4 <= -0.000977]
"Ind_ 201"  "Ind_ 202"  "Ind_ 203"  "Ind_ 204"
"Ind_ 205"  "Ind_ 206"  "Ind_ 207"  "Ind_ 208"
"Ind_ 209"  "Ind_ 210"  "Ind_ 211"  "Ind_ 212"
"Ind_ 213"  "Ind_ 214"  "Ind_ 215"  "Ind_ 216"
"Ind_ 217"  "Ind_ 218"  "Ind_ 219"  "Ind_ 220"
"Ind_ 221"  "Ind_ 222"  "Ind_ 223"  "Ind_ 224"
"Ind_ 225"  "Ind_ 226"  "Ind_ 227"  "Ind_ 228"
"Ind_ 229"  "Ind_ 230"  "Ind_ 231"  "Ind_ 232"
"Ind_ 233"  "Ind_ 234"  "Ind_ 235"  "Ind_ 236"
"Ind_ 237"  "Ind_ 238"  "Ind_ 239"  "Ind_ 240"
"Ind_ 241"  "Ind_ 242"  "Ind_ 243"  "Ind_ 244"
"Ind_ 245"  "Ind_ 246"  "Ind_ 247"  "Ind_ 248"
"Ind_ 249"  "Ind_ 250"  "Ind_ 251"  "Ind_ 252"
"Ind_ 253"  "Ind_ 254"  "Ind_ 255"  "Ind_ 256"
"Ind_ 257"  "Ind_ 258"  "Ind_ 259"  "Ind_ 260"
"Ind_ 261"  "Ind_ 262"  "Ind_ 263"  "Ind_ 264"
"Ind_ 265"  "Ind_ 266"  "Ind_ 267"  "Ind_ 268"
"Ind_ 269"  "Ind_ 270"  "Ind_ 271"  "Ind_ 272"
"Ind_ 273"  "Ind_ 274"  "Ind_ 275"  "Ind_ 276"
"Ind_ 277"  "Ind_ 278"  "Ind_ 279"  "Ind_ 280"
"Ind_ 281"  "Ind_ 282"  "Ind_ 283"  "Ind_ 284"
"Ind_ 285"  "Ind_ 286"  "Ind_ 287"  "Ind_ 288"
"Ind_ 289"  "Ind_ 290"  "Ind_ 291"  "Ind_ 292"
"Ind_ 293"  "Ind_ 294"  "Ind_ 295"  "Ind_ 296"
"Ind_ 297"  "Ind_ 298"  "Ind_ 299"  "Ind_ 300"

Cluster 3 (Cardinal=100) : [Variable4 > -0.000977]
"Ind_ 1"    "Ind_ 2"    "Ind_ 3"    "Ind_ 4"
"Ind_ 5"    "Ind_ 6"    "Ind_ 7"    "Ind_ 8"

```

"Ind_ 9"	"Ind_ 10"	"Ind_ 11"	"Ind_ 12"
"Ind_ 13"	"Ind_ 14"	"Ind_ 15"	"Ind_ 16"
"Ind_ 17"	"Ind_ 18"	"Ind_ 19"	"Ind_ 20"
"Ind_ 21"	"Ind_ 22"	"Ind_ 23"	"Ind_ 24"
"Ind_ 25"	"Ind_ 26"	"Ind_ 27"	"Ind_ 28"
"Ind_ 29"	"Ind_ 30"	"Ind_ 31"	"Ind_ 32"
"Ind_ 33"	"Ind_ 34"	"Ind_ 35"	"Ind_ 36"
"Ind_ 37"	"Ind_ 38"	"Ind_ 39"	"Ind_ 40"
"Ind_ 41"	"Ind_ 42"	"Ind_ 43"	"Ind_ 44"
"Ind_ 45"	"Ind_ 46"	"Ind_ 47"	"Ind_ 48"
"Ind_ 49"	"Ind_ 50"	"Ind_ 51"	"Ind_ 52"
"Ind_ 53"	"Ind_ 54"	"Ind_ 55"	"Ind_ 56"
"Ind_ 57"	"Ind_ 58"	"Ind_ 59"	"Ind_ 60"
"Ind_ 61"	"Ind_ 62"	"Ind_ 63"	"Ind_ 64"
"Ind_ 65"	"Ind_ 66"	"Ind_ 67"	"Ind_ 68"
"Ind_ 69"	"Ind_ 70"	"Ind_ 71"	"Ind_ 72"
"Ind_ 73"	"Ind_ 74"	"Ind_ 75"	"Ind_ 76"
"Ind_ 77"	"Ind_ 78"	"Ind_ 79"	"Ind_ 80"
"Ind_ 81"	"Ind_ 82"	"Ind_ 83"	"Ind_ 84"
"Ind_ 85"	"Ind_ 86"	"Ind_ 87"	"Ind_ 88"
"Ind_ 89"	"Ind_ 90"	"Ind_ 91"	"Ind_ 92"
"Ind_ 93"	"Ind_ 94"	"Ind_ 95"	"Ind_ 96"
"Ind_ 97"	"Ind_ 98"	"Ind_ 99"	"Ind_ 100"

Cluster 4 (Cardinal=100) : [Variable4 > -0.000824]

"Ind_ 101"	"Ind_ 102"	"Ind_ 103"	"Ind_ 104"
"Ind_ 105"	"Ind_ 106"	"Ind_ 107"	"Ind_ 108"
"Ind_ 109"	"Ind_ 110"	"Ind_ 111"	"Ind_ 112"
"Ind_ 113"	"Ind_ 114"	"Ind_ 115"	"Ind_ 116"
"Ind_ 117"	"Ind_ 118"	"Ind_ 119"	"Ind_ 120"
"Ind_ 121"	"Ind_ 122"	"Ind_ 123"	"Ind_ 124"
"Ind_ 125"	"Ind_ 126"	"Ind_ 127"	"Ind_ 128"
"Ind_ 129"	"Ind_ 130"	"Ind_ 131"	"Ind_ 132"
"Ind_ 133"	"Ind_ 134"	"Ind_ 135"	"Ind_ 136"
"Ind_ 137"	"Ind_ 138"	"Ind_ 139"	"Ind_ 140"
"Ind_ 141"	"Ind_ 142"	"Ind_ 143"	"Ind_ 144"
"Ind_ 145"	"Ind_ 146"	"Ind_ 147"	"Ind_ 148"
"Ind_ 149"	"Ind_ 150"	"Ind_ 151"	"Ind_ 152"
"Ind_ 153"	"Ind_ 154"	"Ind_ 155"	"Ind_ 156"
"Ind_ 157"	"Ind_ 158"	"Ind_ 159"	"Ind_ 160"
"Ind_ 161"	"Ind_ 162"	"Ind_ 163"	"Ind_ 164"
"Ind_ 165"	"Ind_ 166"	"Ind_ 167"	"Ind_ 168"
"Ind_ 169"	"Ind_ 170"	"Ind_ 171"	"Ind_ 172"
"Ind_ 173"	"Ind_ 174"	"Ind_ 175"	"Ind_ 176"
"Ind_ 177"	"Ind_ 178"	"Ind_ 179"	"Ind_ 180"
"Ind_ 181"	"Ind_ 182"	"Ind_ 183"	"Ind_ 184"
"Ind_ 185"	"Ind_ 186"	"Ind_ 187"	"Ind_ 188"
"Ind_ 189"	"Ind_ 190"	"Ind_ 191"	"Ind_ 192"
"Ind_ 193"	"Ind_ 194"	"Ind_ 195"	"Ind_ 196"
"Ind_ 197"	"Ind_ 198"	"Ind_ 199"	"Ind_ 200"

Inertie expliquée : 74.064706

PARTITION IN 5 CLUSTERS :

Cluster 1 (Cardinal=55) : [Variable1 <= 1.036270]

"Ind_ 301"	"Ind_ 302"	"Ind_ 304"	"Ind_ 310"
"Ind_ 312"	"Ind_ 313"	"Ind_ 317"	"Ind_ 318"
"Ind_ 319"	"Ind_ 320"	"Ind_ 321"	"Ind_ 322"
"Ind_ 323"	"Ind_ 325"	"Ind_ 329"	"Ind_ 331"
"Ind_ 333"	"Ind_ 334"	"Ind_ 336"	"Ind_ 337"
"Ind_ 338"	"Ind_ 339"	"Ind_ 341"	"Ind_ 345"
"Ind_ 347"	"Ind_ 348"	"Ind_ 349"	"Ind_ 350"
"Ind_ 351"	"Ind_ 352"	"Ind_ 354"	"Ind_ 358"
"Ind_ 359"	"Ind_ 360"	"Ind_ 361"	"Ind_ 362"
"Ind_ 363"	"Ind_ 366"	"Ind_ 367"	"Ind_ 370"

"Ind_ 372"	"Ind_ 373"	"Ind_ 374"	"Ind_ 375"
"Ind_ 376"	"Ind_ 379"	"Ind_ 381"	"Ind_ 382"
"Ind_ 386"	"Ind_ 386"	"Ind_ 389"	"Ind_ 390"
"Ind_ 394"	"Ind_ 399"	"Ind_ 400"	

Cluster 2 (Cardinal=100) : [Variable4 <= -0.000977]

"Ind_ 201"	"Ind_ 202"	"Ind_ 203"	"Ind_ 204"
"Ind_ 205"	"Ind_ 206"	"Ind_ 207"	"Ind_ 208"
"Ind_ 209"	"Ind_ 210"	"Ind_ 211"	"Ind_ 212"
"Ind_ 213"	"Ind_ 214"	"Ind_ 215"	"Ind_ 216"
"Ind_ 217"	"Ind_ 218"	"Ind_ 219"	"Ind_ 220"
"Ind_ 221"	"Ind_ 222"	"Ind_ 223"	"Ind_ 224"
"Ind_ 225"	"Ind_ 226"	"Ind_ 227"	"Ind_ 228"
"Ind_ 229"	"Ind_ 230"	"Ind_ 231"	"Ind_ 232"
"Ind_ 233"	"Ind_ 234"	"Ind_ 235"	"Ind_ 236"
"Ind_ 237"	"Ind_ 238"	"Ind_ 239"	"Ind_ 240"
"Ind_ 241"	"Ind_ 242"	"Ind_ 243"	"Ind_ 244"
"Ind_ 245"	"Ind_ 246"	"Ind_ 247"	"Ind_ 248"
"Ind_ 249"	"Ind_ 250"	"Ind_ 251"	"Ind_ 252"
"Ind_ 253"	"Ind_ 254"	"Ind_ 255"	"Ind_ 256"
"Ind_ 257"	"Ind_ 258"	"Ind_ 259"	"Ind_ 260"
"Ind_ 261"	"Ind_ 262"	"Ind_ 263"	"Ind_ 264"
"Ind_ 265"	"Ind_ 266"	"Ind_ 267"	"Ind_ 268"
"Ind_ 269"	"Ind_ 270"	"Ind_ 271"	"Ind_ 272"
"Ind_ 273"	"Ind_ 274"	"Ind_ 275"	"Ind_ 276"
"Ind_ 277"	"Ind_ 278"	"Ind_ 279"	"Ind_ 280"
"Ind_ 281"	"Ind_ 282"	"Ind_ 283"	"Ind_ 284"
"Ind_ 285"	"Ind_ 286"	"Ind_ 287"	"Ind_ 288"
"Ind_ 289"	"Ind_ 290"	"Ind_ 291"	"Ind_ 292"
"Ind_ 293"	"Ind_ 294"	"Ind_ 295"	"Ind_ 296"
"Ind_ 297"	"Ind_ 298"	"Ind_ 299"	"Ind_ 300"

Cluster 3 (Cardinal=100) : [Variable4 > -0.000977]

"Ind_ 1"	"Ind_ 2"	"Ind_ 3"	"Ind_ 4"
"Ind_ 5"	"Ind_ 6"	"Ind_ 7"	"Ind_ 8"
"Ind_ 9"	"Ind_ 10"	"Ind_ 11"	"Ind_ 12"
"Ind_ 13"	"Ind_ 14"	"Ind_ 15"	"Ind_ 16"
"Ind_ 17"	"Ind_ 18"	"Ind_ 19"	"Ind_ 20"
"Ind_ 21"	"Ind_ 22"	"Ind_ 23"	"Ind_ 24"
"Ind_ 25"	"Ind_ 26"	"Ind_ 27"	"Ind_ 28"
"Ind_ 29"	"Ind_ 30"	"Ind_ 31"	"Ind_ 32"
"Ind_ 33"	"Ind_ 34"	"Ind_ 35"	"Ind_ 36"
"Ind_ 37"	"Ind_ 38"	"Ind_ 39"	"Ind_ 40"
"Ind_ 41"	"Ind_ 42"	"Ind_ 43"	"Ind_ 44"
"Ind_ 45"	"Ind_ 46"	"Ind_ 47"	"Ind_ 48"
"Ind_ 49"	"Ind_ 50"	"Ind_ 51"	"Ind_ 52"
"Ind_ 53"	"Ind_ 54"	"Ind_ 55"	"Ind_ 56"
"Ind_ 57"	"Ind_ 58"	"Ind_ 59"	"Ind_ 60"
"Ind_ 61"	"Ind_ 62"	"Ind_ 63"	"Ind_ 64"
"Ind_ 65"	"Ind_ 66"	"Ind_ 67"	"Ind_ 68"
"Ind_ 69"	"Ind_ 70"	"Ind_ 71"	"Ind_ 72"
"Ind_ 73"	"Ind_ 74"	"Ind_ 75"	"Ind_ 76"
"Ind_ 77"	"Ind_ 78"	"Ind_ 79"	"Ind_ 80"
"Ind_ 81"	"Ind_ 82"	"Ind_ 83"	"Ind_ 84"
"Ind_ 85"	"Ind_ 86"	"Ind_ 87"	"Ind_ 88"
"Ind_ 89"	"Ind_ 90"	"Ind_ 91"	"Ind_ 92"
"Ind_ 93"	"Ind_ 94"	"Ind_ 95"	"Ind_ 96"
"Ind_ 97"	"Ind_ 98"	"Ind_ 99"	"Ind_ 100"

Cluster 4 (Cardinal=100) : [Variable4 > -0.000824]

"Ind_ 101"	"Ind_ 102"	"Ind_ 103"	"Ind_ 104"
"Ind_ 105"	"Ind_ 106"	"Ind_ 107"	"Ind_ 108"
"Ind_ 109"	"Ind_ 110"	"Ind_ 111"	"Ind_ 112"
"Ind_ 113"	"Ind_ 114"	"Ind_ 115"	"Ind_ 116"
"Ind_ 117"	"Ind_ 118"	"Ind_ 119"	"Ind_ 120"
"Ind_ 121"	"Ind_ 122"	"Ind_ 123"	"Ind_ 124"
"Ind_ 125"	"Ind_ 126"	"Ind_ 127"	"Ind_ 128"

```

"Ind_ 129"    "Ind_ 130"    "Ind_ 131"    "Ind_ 132"
"Ind_ 133"    "Ind_ 134"    "Ind_ 135"    "Ind_ 136"
"Ind_ 137"    "Ind_ 138"    "Ind_ 139"    "Ind_ 140"
"Ind_ 141"    "Ind_ 142"    "Ind_ 143"    "Ind_ 144"
"Ind_ 145"    "Ind_ 146"    "Ind_ 147"    "Ind_ 148"
"Ind_ 149"    "Ind_ 150"    "Ind_ 151"    "Ind_ 152"
"Ind_ 153"    "Ind_ 154"    "Ind_ 155"    "Ind_ 156"
"Ind_ 157"    "Ind_ 158"    "Ind_ 159"    "Ind_ 160"
"Ind_ 161"    "Ind_ 162"    "Ind_ 163"    "Ind_ 164"
"Ind_ 165"    "Ind_ 166"    "Ind_ 167"    "Ind_ 168"
"Ind_ 169"    "Ind_ 170"    "Ind_ 171"    "Ind_ 172"
"Ind_ 173"    "Ind_ 174"    "Ind_ 175"    "Ind_ 176"
"Ind_ 177"    "Ind_ 178"    "Ind_ 179"    "Ind_ 180"
"Ind_ 181"    "Ind_ 182"    "Ind_ 183"    "Ind_ 184"
"Ind_ 185"    "Ind_ 186"    "Ind_ 187"    "Ind_ 188"
"Ind_ 189"    "Ind_ 190"    "Ind_ 191"    "Ind_ 192"
"Ind_ 193"    "Ind_ 194"    "Ind_ 195"    "Ind_ 196"
"Ind_ 197"    "Ind_ 198"    "Ind_ 199"    "Ind_ 200"

```

```

Cluster 5 (Cardinal=45) : [Variable1 > 1.036270]
"Ind_ 303"    "Ind_ 305"    "Ind_ 306"    "Ind_ 307"
"Ind_ 308"    "Ind_ 309"    "Ind_ 311"    "Ind_ 314"
"Ind_ 315"    "Ind_ 316"    "Ind_ 324"    "Ind_ 326"
"Ind_ 327"    "Ind_ 328"    "Ind_ 330"    "Ind_ 332"
"Ind_ 335"    "Ind_ 340"    "Ind_ 342"    "Ind_ 343"
"Ind_ 344"    "Ind_ 346"    "Ind_ 353"    "Ind_ 355"
"Ind_ 356"    "Ind_ 357"    "Ind_ 364"    "Ind_ 365"
"Ind_ 368"    "Ind_ 369"    "Ind_ 371"    "Ind_ 376"
"Ind_ 377"    "Ind_ 380"    "Ind_ 383"    "Ind_ 384"
"Ind_ 387"    "Ind_ 388"    "Ind_ 391"    "Ind_ 392"
"Ind_ 393"    "Ind_ 395"    "Ind_ 396"    "Ind_ 397"
"Ind_ 398"

```

Inertie expliquée : 75.047324

THE CLUSTERING TREE :

-
- the number noted at each node indicates the order of the division
 - Ng <-> yes and Nd <-> no

```

          +---- Classe 1 (Ng=55)
          !
          !----4- [Variable1 <= 1.036270]
          !
          !
          ! +---- Classe 5 (Nd=45)
          !
          !----3- [Variable4 <= -0.000824]
          !
          !
          ! +---- Classe 4 (Nd=100)
          !
          !----1- [Variable5 <= -0.003006]
          !
          !
          ! +---- Classe 2 (Ng=100)
          !
          !----2- [Variable4 <= -0.000977]
          !
          !
          ! +---- Classe 3 (Nd=100)

```

Si nous faisons un graphique de l'évolution de l'inertie expliquée, nous obtenons :

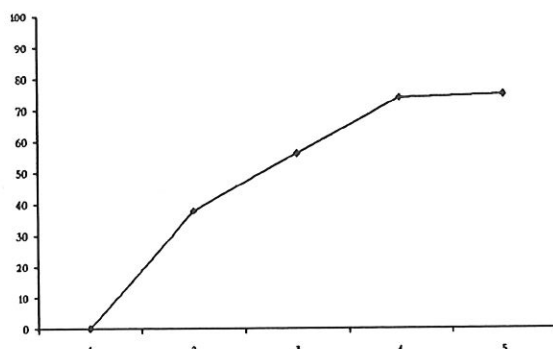


FIGURE 13. Evolution de l'inertie expliquée en fonction du nombre de classes

Le nombre de classes naturelles que nous pouvons distinguer est donc bien 4 comme nous l'avions décidé avant de créer le jeu de données, l'inertie expliquée par le passage de 4 classes à 5 classes étant négligeable.

Nous regroupons ces résultats dans un fichier de sortie en vue de la comparaison : `chavent.txt` (Annexe D).

11.2.2. Méthode des nuées dynamiques

De la même manière, nous exécutons le programme en choisissant la méthode des nuées dynamiques DYCLASS et le calcul des classifications successives de 2 à 5 classes. Nous obtenons les résultats suivants, extraits du fichier `artificielles.txt` :

PARTITION IN 2 CLUSTERS :

Cluster 1 (Cardinal=200) :

"Ind_ 101"	"Ind_ 102"	"Ind_ 103"	"Ind_ 104"
"Ind_ 105"	"Ind_ 106"	"Ind_ 107"	"Ind_ 108"
"Ind_ 109"	"Ind_ 110"	"Ind_ 111"	"Ind_ 112"
"Ind_ 113"	"Ind_ 114"	"Ind_ 115"	"Ind_ 116"
"Ind_ 117"	"Ind_ 118"	"Ind_ 119"	"Ind_ 120"
"Ind_ 121"	"Ind_ 122"	"Ind_ 123"	"Ind_ 124"
"Ind_ 125"	"Ind_ 126"	"Ind_ 127"	"Ind_ 128"
"Ind_ 129"	"Ind_ 130"	"Ind_ 131"	"Ind_ 132"
"Ind_ 133"	"Ind_ 134"	"Ind_ 135"	"Ind_ 136"
"Ind_ 137"	"Ind_ 138"	"Ind_ 139"	"Ind_ 140"
"Ind_ 141"	"Ind_ 142"	"Ind_ 143"	"Ind_ 144"
"Ind_ 145"	"Ind_ 146"	"Ind_ 147"	"Ind_ 148"
"Ind_ 149"	"Ind_ 150"	"Ind_ 151"	"Ind_ 152"
"Ind_ 153"	"Ind_ 154"	"Ind_ 155"	"Ind_ 156"

"Ind_ 157"	"Ind_ 158"	"Ind_ 159"	"Ind_ 160"
"Ind_ 161"	"Ind_ 162"	"Ind_ 163"	"Ind_ 164"
"Ind_ 165"	"Ind_ 166"	"Ind_ 167"	"Ind_ 168"
"Ind_ 169"	"Ind_ 170"	"Ind_ 171"	"Ind_ 172"
"Ind_ 173"	"Ind_ 174"	"Ind_ 175"	"Ind_ 176"
"Ind_ 177"	"Ind_ 178"	"Ind_ 179"	"Ind_ 180"
"Ind_ 181"	"Ind_ 182"	"Ind_ 183"	"Ind_ 184"
"Ind_ 185"	"Ind_ 186"	"Ind_ 187"	"Ind_ 188"
"Ind_ 189"	"Ind_ 190"	"Ind_ 191"	"Ind_ 192"
"Ind_ 193"	"Ind_ 194"	"Ind_ 195"	"Ind_ 196"
"Ind_ 197"	"Ind_ 198"	"Ind_ 199"	"Ind_ 200"
"Ind_ 301"	"Ind_ 302"	"Ind_ 303"	"Ind_ 304"
"Ind_ 305"	"Ind_ 306"	"Ind_ 307"	"Ind_ 308"
"Ind_ 309"	"Ind_ 310"	"Ind_ 311"	"Ind_ 312"
"Ind_ 313"	"Ind_ 314"	"Ind_ 315"	"Ind_ 316"
"Ind_ 317"	"Ind_ 318"	"Ind_ 319"	"Ind_ 320"
"Ind_ 321"	"Ind_ 322"	"Ind_ 323"	"Ind_ 324"
"Ind_ 325"	"Ind_ 326"	"Ind_ 327"	"Ind_ 328"
"Ind_ 329"	"Ind_ 330"	"Ind_ 331"	"Ind_ 332"
"Ind_ 333"	"Ind_ 334"	"Ind_ 335"	"Ind_ 336"
"Ind_ 337"	"Ind_ 338"	"Ind_ 339"	"Ind_ 340"
"Ind_ 341"	"Ind_ 342"	"Ind_ 343"	"Ind_ 344"
"Ind_ 345"	"Ind_ 346"	"Ind_ 347"	"Ind_ 348"
"Ind_ 349"	"Ind_ 350"	"Ind_ 351"	"Ind_ 352"
"Ind_ 353"	"Ind_ 354"	"Ind_ 355"	"Ind_ 356"
"Ind_ 357"	"Ind_ 358"	"Ind_ 359"	"Ind_ 360"
"Ind_ 361"	"Ind_ 362"	"Ind_ 363"	"Ind_ 364"
"Ind_ 365"	"Ind_ 366"	"Ind_ 367"	"Ind_ 368"
"Ind_ 369"	"Ind_ 370"	"Ind_ 371"	"Ind_ 372"
"Ind_ 373"	"Ind_ 374"	"Ind_ 375"	"Ind_ 376"
"Ind_ 377"	"Ind_ 378"	"Ind_ 379"	"Ind_ 380"
"Ind_ 381"	"Ind_ 382"	"Ind_ 383"	"Ind_ 384"
"Ind_ 385"	"Ind_ 386"	"Ind_ 387"	"Ind_ 388"
"Ind_ 389"	"Ind_ 390"	"Ind_ 391"	"Ind_ 392"
"Ind_ 393"	"Ind_ 394"	"Ind_ 395"	"Ind_ 396"
"Ind_ 397"	"Ind_ 398"	"Ind_ 399"	"Ind_ 400"

Cluster 2 (Cardinal=200) :

"Ind_ 1"	"Ind_ 2"	"Ind_ 3"	"Ind_ 4"
"Ind_ 5"	"Ind_ 6"	"Ind_ 7"	"Ind_ 8"
"Ind_ 9"	"Ind_ 10"	"Ind_ 11"	"Ind_ 12"
"Ind_ 13"	"Ind_ 14"	"Ind_ 15"	"Ind_ 16"
"Ind_ 17"	"Ind_ 18"	"Ind_ 19"	"Ind_ 20"
"Ind_ 21"	"Ind_ 22"	"Ind_ 23"	"Ind_ 24"
"Ind_ 25"	"Ind_ 26"	"Ind_ 27"	"Ind_ 28"
"Ind_ 29"	"Ind_ 30"	"Ind_ 31"	"Ind_ 32"
"Ind_ 33"	"Ind_ 34"	"Ind_ 35"	"Ind_ 36"
"Ind_ 37"	"Ind_ 38"	"Ind_ 39"	"Ind_ 40"
"Ind_ 41"	"Ind_ 42"	"Ind_ 43"	"Ind_ 44"
"Ind_ 45"	"Ind_ 46"	"Ind_ 47"	"Ind_ 48"
"Ind_ 49"	"Ind_ 50"	"Ind_ 51"	"Ind_ 52"
"Ind_ 53"	"Ind_ 54"	"Ind_ 55"	"Ind_ 56"
"Ind_ 57"	"Ind_ 58"	"Ind_ 59"	"Ind_ 60"
"Ind_ 61"	"Ind_ 62"	"Ind_ 63"	"Ind_ 64"
"Ind_ 65"	"Ind_ 66"	"Ind_ 67"	"Ind_ 68"
"Ind_ 69"	"Ind_ 70"	"Ind_ 71"	"Ind_ 72"
"Ind_ 73"	"Ind_ 74"	"Ind_ 75"	"Ind_ 76"
"Ind_ 77"	"Ind_ 78"	"Ind_ 79"	"Ind_ 80"
"Ind_ 81"	"Ind_ 82"	"Ind_ 83"	"Ind_ 84"
"Ind_ 85"	"Ind_ 86"	"Ind_ 87"	"Ind_ 88"
"Ind_ 89"	"Ind_ 90"	"Ind_ 91"	"Ind_ 92"
"Ind_ 93"	"Ind_ 94"	"Ind_ 95"	"Ind_ 96"
"Ind_ 97"	"Ind_ 98"	"Ind_ 99"	"Ind_ 100"
"Ind_ 201"	"Ind_ 202"	"Ind_ 203"	"Ind_ 204"
"Ind_ 205"	"Ind_ 206"	"Ind_ 207"	"Ind_ 208"
"Ind_ 209"	"Ind_ 210"	"Ind_ 211"	"Ind_ 212"
"Ind_ 213"	"Ind_ 214"	"Ind_ 215"	"Ind_ 216"

"Ind_ 217"	"Ind_ 218"	"Ind_ 219"	"Ind_ 220"
"Ind_ 221"	"Ind_ 222"	"Ind_ 223"	"Ind_ 224"
"Ind_ 225"	"Ind_ 226"	"Ind_ 227"	"Ind_ 228"
"Ind_ 229"	"Ind_ 230"	"Ind_ 231"	"Ind_ 232"
"Ind_ 233"	"Ind_ 234"	"Ind_ 235"	"Ind_ 236"
"Ind_ 237"	"Ind_ 238"	"Ind_ 239"	"Ind_ 240"
"Ind_ 241"	"Ind_ 242"	"Ind_ 243"	"Ind_ 244"
"Ind_ 245"	"Ind_ 246"	"Ind_ 247"	"Ind_ 248"
"Ind_ 249"	"Ind_ 250"	"Ind_ 251"	"Ind_ 252"
"Ind_ 253"	"Ind_ 254"	"Ind_ 255"	"Ind_ 256"
"Ind_ 257"	"Ind_ 258"	"Ind_ 259"	"Ind_ 260"
"Ind_ 261"	"Ind_ 262"	"Ind_ 263"	"Ind_ 264"
"Ind_ 265"	"Ind_ 266"	"Ind_ 267"	"Ind_ 268"
"Ind_ 269"	"Ind_ 270"	"Ind_ 271"	"Ind_ 272"
"Ind_ 273"	"Ind_ 274"	"Ind_ 275"	"Ind_ 276"
"Ind_ 277"	"Ind_ 278"	"Ind_ 279"	"Ind_ 280"
"Ind_ 281"	"Ind_ 282"	"Ind_ 283"	"Ind_ 284"
"Ind_ 285"	"Ind_ 286"	"Ind_ 287"	"Ind_ 288"
"Ind_ 289"	"Ind_ 290"	"Ind_ 291"	"Ind_ 292"
"Ind_ 293"	"Ind_ 294"	"Ind_ 295"	"Ind_ 296"
"Ind_ 297"	"Ind_ 298"	"Ind_ 299"	"Ind_ 300"

Inertie expliquée : 37.702664

PARTITION IN 3 CLUSTERS :

Cluster 1 (Cardinal=100) :

"Ind_ 1"	"Ind_ 2"	"Ind_ 3"	"Ind_ 4"
"Ind_ 5"	"Ind_ 6"	"Ind_ 7"	"Ind_ 8"
"Ind_ 9"	"Ind_ 10"	"Ind_ 11"	"Ind_ 12"
"Ind_ 13"	"Ind_ 14"	"Ind_ 15"	"Ind_ 16"
"Ind_ 17"	"Ind_ 18"	"Ind_ 19"	"Ind_ 20"
"Ind_ 21"	"Ind_ 22"	"Ind_ 23"	"Ind_ 24"
"Ind_ 25"	"Ind_ 26"	"Ind_ 27"	"Ind_ 28"
"Ind_ 29"	"Ind_ 30"	"Ind_ 31"	"Ind_ 32"
"Ind_ 33"	"Ind_ 34"	"Ind_ 35"	"Ind_ 36"
"Ind_ 37"	"Ind_ 38"	"Ind_ 39"	"Ind_ 40"
"Ind_ 41"	"Ind_ 42"	"Ind_ 43"	"Ind_ 44"
"Ind_ 45"	"Ind_ 46"	"Ind_ 47"	"Ind_ 48"
"Ind_ 49"	"Ind_ 50"	"Ind_ 51"	"Ind_ 52"
"Ind_ 53"	"Ind_ 54"	"Ind_ 55"	"Ind_ 56"
"Ind_ 57"	"Ind_ 58"	"Ind_ 59"	"Ind_ 60"
"Ind_ 61"	"Ind_ 62"	"Ind_ 63"	"Ind_ 64"
"Ind_ 65"	"Ind_ 66"	"Ind_ 67"	"Ind_ 68"
"Ind_ 69"	"Ind_ 70"	"Ind_ 71"	"Ind_ 72"
"Ind_ 73"	"Ind_ 74"	"Ind_ 75"	"Ind_ 76"
"Ind_ 77"	"Ind_ 78"	"Ind_ 79"	"Ind_ 80"
"Ind_ 81"	"Ind_ 82"	"Ind_ 83"	"Ind_ 84"
"Ind_ 85"	"Ind_ 86"	"Ind_ 87"	"Ind_ 88"
"Ind_ 89"	"Ind_ 90"	"Ind_ 91"	"Ind_ 92"
"Ind_ 93"	"Ind_ 94"	"Ind_ 95"	"Ind_ 96"
"Ind_ 97"	"Ind_ 98"	"Ind_ 99"	"Ind_ 100"

Cluster 2 (Cardinal=177) :

"Ind_ 101"	"Ind_ 102"	"Ind_ 103"	"Ind_ 104"
"Ind_ 105"	"Ind_ 106"	"Ind_ 107"	"Ind_ 108"
"Ind_ 109"	"Ind_ 110"	"Ind_ 111"	"Ind_ 112"
"Ind_ 113"	"Ind_ 114"	"Ind_ 115"	"Ind_ 116"
"Ind_ 117"	"Ind_ 118"	"Ind_ 119"	"Ind_ 120"
"Ind_ 121"	"Ind_ 122"	"Ind_ 123"	"Ind_ 124"
"Ind_ 125"	"Ind_ 126"	"Ind_ 127"	"Ind_ 128"
"Ind_ 129"	"Ind_ 130"	"Ind_ 131"	"Ind_ 132"
"Ind_ 133"	"Ind_ 134"	"Ind_ 135"	"Ind_ 136"
"Ind_ 137"	"Ind_ 138"	"Ind_ 139"	"Ind_ 140"
"Ind_ 141"	"Ind_ 142"	"Ind_ 143"	"Ind_ 144"

"Ind_ 145"	"Ind_ 146"	"Ind_ 147"	"Ind_ 148"
"Ind_ 149"	"Ind_ 150"	"Ind_ 151"	"Ind_ 152"
"Ind_ 153"	"Ind_ 154"	"Ind_ 155"	"Ind_ 156"
"Ind_ 157"	"Ind_ 158"	"Ind_ 159"	"Ind_ 160"
"Ind_ 161"	"Ind_ 162"	"Ind_ 163"	"Ind_ 164"
"Ind_ 165"	"Ind_ 166"	"Ind_ 167"	"Ind_ 168"
"Ind_ 169"	"Ind_ 170"	"Ind_ 171"	"Ind_ 172"
"Ind_ 173"	"Ind_ 174"	"Ind_ 175"	"Ind_ 176"
"Ind_ 177"	"Ind_ 178"	"Ind_ 179"	"Ind_ 180"
"Ind_ 181"	"Ind_ 182"	"Ind_ 183"	"Ind_ 184"
"Ind_ 185"	"Ind_ 186"	"Ind_ 187"	"Ind_ 188"
"Ind_ 189"	"Ind_ 190"	"Ind_ 191"	"Ind_ 192"
"Ind_ 193"	"Ind_ 194"	"Ind_ 195"	"Ind_ 196"
"Ind_ 197"	"Ind_ 198"	"Ind_ 199"	"Ind_ 200"
"Ind_ 302"	"Ind_ 303"	"Ind_ 304"	"Ind_ 305"
"Ind_ 306"	"Ind_ 309"	"Ind_ 310"	"Ind_ 312"
"Ind_ 313"	"Ind_ 314"	"Ind_ 315"	"Ind_ 316"
"Ind_ 317"	"Ind_ 320"	"Ind_ 322"	"Ind_ 323"
"Ind_ 324"	"Ind_ 326"	"Ind_ 327"	"Ind_ 328"
"Ind_ 329"	"Ind_ 331"	"Ind_ 332"	"Ind_ 334"
"Ind_ 335"	"Ind_ 336"	"Ind_ 337"	"Ind_ 338"
"Ind_ 340"	"Ind_ 341"	"Ind_ 342"	"Ind_ 343"
"Ind_ 344"	"Ind_ 345"	"Ind_ 346"	"Ind_ 347"
"Ind_ 348"	"Ind_ 349"	"Ind_ 350"	"Ind_ 351"
"Ind_ 352"	"Ind_ 353"	"Ind_ 355"	"Ind_ 356"
"Ind_ 357"	"Ind_ 358"	"Ind_ 359"	"Ind_ 361"
"Ind_ 362"	"Ind_ 363"	"Ind_ 364"	"Ind_ 365"
"Ind_ 366"	"Ind_ 367"	"Ind_ 368"	"Ind_ 369"
"Ind_ 370"	"Ind_ 374"	"Ind_ 375"	"Ind_ 376"
"Ind_ 377"	"Ind_ 378"	"Ind_ 382"	"Ind_ 383"
"Ind_ 384"	"Ind_ 385"	"Ind_ 387"	"Ind_ 388"
"Ind_ 389"	"Ind_ 391"	"Ind_ 392"	"Ind_ 393"
"Ind_ 394"	"Ind_ 395"	"Ind_ 397"	"Ind_ 399"
"Ind_ 400"			

Cluster 3 (Cardinal=123) :

"Ind_ 201"	"Ind_ 202"	"Ind_ 203"	"Ind_ 204"
"Ind_ 205"	"Ind_ 206"	"Ind_ 207"	"Ind_ 208"
"Ind_ 209"	"Ind_ 210"	"Ind_ 211"	"Ind_ 212"
"Ind_ 213"	"Ind_ 214"	"Ind_ 215"	"Ind_ 216"
"Ind_ 217"	"Ind_ 218"	"Ind_ 219"	"Ind_ 220"
"Ind_ 221"	"Ind_ 222"	"Ind_ 223"	"Ind_ 224"
"Ind_ 225"	"Ind_ 226"	"Ind_ 227"	"Ind_ 228"
"Ind_ 229"	"Ind_ 230"	"Ind_ 231"	"Ind_ 232"
"Ind_ 233"	"Ind_ 234"	"Ind_ 235"	"Ind_ 236"
"Ind_ 237"	"Ind_ 238"	"Ind_ 239"	"Ind_ 240"
"Ind_ 241"	"Ind_ 242"	"Ind_ 243"	"Ind_ 244"
"Ind_ 245"	"Ind_ 246"	"Ind_ 247"	"Ind_ 248"
"Ind_ 249"	"Ind_ 250"	"Ind_ 251"	"Ind_ 252"
"Ind_ 253"	"Ind_ 254"	"Ind_ 255"	"Ind_ 256"
"Ind_ 257"	"Ind_ 258"	"Ind_ 259"	"Ind_ 260"
"Ind_ 261"	"Ind_ 262"	"Ind_ 263"	"Ind_ 264"
"Ind_ 265"	"Ind_ 266"	"Ind_ 267"	"Ind_ 268"
"Ind_ 269"	"Ind_ 270"	"Ind_ 271"	"Ind_ 272"
"Ind_ 273"	"Ind_ 274"	"Ind_ 275"	"Ind_ 276"
"Ind_ 277"	"Ind_ 278"	"Ind_ 279"	"Ind_ 280"
"Ind_ 281"	"Ind_ 282"	"Ind_ 283"	"Ind_ 284"
"Ind_ 285"	"Ind_ 286"	"Ind_ 287"	"Ind_ 288"
"Ind_ 289"	"Ind_ 290"	"Ind_ 291"	"Ind_ 292"
"Ind_ 293"	"Ind_ 294"	"Ind_ 295"	"Ind_ 296"
"Ind_ 297"	"Ind_ 298"	"Ind_ 299"	"Ind_ 300"
"Ind_ 301"	"Ind_ 307"	"Ind_ 308"	"Ind_ 311"
"Ind_ 318"	"Ind_ 319"	"Ind_ 321"	"Ind_ 325"
"Ind_ 330"	"Ind_ 333"	"Ind_ 339"	"Ind_ 354"
"Ind_ 360"	"Ind_ 371"	"Ind_ 372"	"Ind_ 373"
"Ind_ 379"	"Ind_ 380"	"Ind_ 381"	"Ind_ 386"
"Ind_ 390"	"Ind_ 396"	"Ind_ 398"	

Inertie expliquée : 52.735275

PARTITION IN 4 CLUSTERS :

Cluster 1 (Cardinal=100) :

"Ind_ 101"	"Ind_ 102"	"Ind_ 103"	"Ind_ 104"
"Ind_ 105"	"Ind_ 106"	"Ind_ 107"	"Ind_ 108"
"Ind_ 109"	"Ind_ 110"	"Ind_ 111"	"Ind_ 112"
"Ind_ 113"	"Ind_ 114"	"Ind_ 115"	"Ind_ 116"
"Ind_ 117"	"Ind_ 118"	"Ind_ 119"	"Ind_ 120"
"Ind_ 121"	"Ind_ 122"	"Ind_ 123"	"Ind_ 124"
"Ind_ 125"	"Ind_ 126"	"Ind_ 127"	"Ind_ 128"
"Ind_ 129"	"Ind_ 130"	"Ind_ 131"	"Ind_ 132"
"Ind_ 133"	"Ind_ 134"	"Ind_ 135"	"Ind_ 136"
"Ind_ 137"	"Ind_ 138"	"Ind_ 139"	"Ind_ 140"
"Ind_ 141"	"Ind_ 142"	"Ind_ 143"	"Ind_ 144"
"Ind_ 145"	"Ind_ 146"	"Ind_ 147"	"Ind_ 148"
"Ind_ 149"	"Ind_ 150"	"Ind_ 151"	"Ind_ 152"
"Ind_ 153"	"Ind_ 154"	"Ind_ 155"	"Ind_ 156"
"Ind_ 157"	"Ind_ 158"	"Ind_ 159"	"Ind_ 160"
"Ind_ 161"	"Ind_ 162"	"Ind_ 163"	"Ind_ 164"
"Ind_ 165"	"Ind_ 166"	"Ind_ 167"	"Ind_ 168"
"Ind_ 169"	"Ind_ 170"	"Ind_ 171"	"Ind_ 172"
"Ind_ 173"	"Ind_ 174"	"Ind_ 175"	"Ind_ 176"
"Ind_ 177"	"Ind_ 178"	"Ind_ 179"	"Ind_ 180"
"Ind_ 181"	"Ind_ 182"	"Ind_ 183"	"Ind_ 184"
"Ind_ 185"	"Ind_ 186"	"Ind_ 187"	"Ind_ 188"
"Ind_ 189"	"Ind_ 190"	"Ind_ 191"	"Ind_ 192"
"Ind_ 193"	"Ind_ 194"	"Ind_ 195"	"Ind_ 196"
"Ind_ 197"	"Ind_ 198"	"Ind_ 199"	"Ind_ 200"

Cluster 2 (Cardinal=100) :

"Ind_ 301"	"Ind_ 302"	"Ind_ 303"	"Ind_ 304"
"Ind_ 305"	"Ind_ 306"	"Ind_ 307"	"Ind_ 308"
"Ind_ 309"	"Ind_ 310"	"Ind_ 311"	"Ind_ 312"
"Ind_ 313"	"Ind_ 314"	"Ind_ 315"	"Ind_ 316"
"Ind_ 317"	"Ind_ 318"	"Ind_ 319"	"Ind_ 320"
"Ind_ 321"	"Ind_ 322"	"Ind_ 323"	"Ind_ 324"
"Ind_ 325"	"Ind_ 326"	"Ind_ 327"	"Ind_ 328"
"Ind_ 329"	"Ind_ 330"	"Ind_ 331"	"Ind_ 332"
"Ind_ 333"	"Ind_ 334"	"Ind_ 335"	"Ind_ 336"
"Ind_ 337"	"Ind_ 338"	"Ind_ 339"	"Ind_ 340"
"Ind_ 341"	"Ind_ 342"	"Ind_ 343"	"Ind_ 344"
"Ind_ 345"	"Ind_ 346"	"Ind_ 347"	"Ind_ 348"
"Ind_ 349"	"Ind_ 350"	"Ind_ 351"	"Ind_ 352"
"Ind_ 353"	"Ind_ 354"	"Ind_ 355"	"Ind_ 356"
"Ind_ 357"	"Ind_ 358"	"Ind_ 359"	"Ind_ 360"
"Ind_ 361"	"Ind_ 362"	"Ind_ 363"	"Ind_ 364"
"Ind_ 365"	"Ind_ 366"	"Ind_ 367"	"Ind_ 368"
"Ind_ 369"	"Ind_ 370"	"Ind_ 371"	"Ind_ 372"
"Ind_ 373"	"Ind_ 374"	"Ind_ 375"	"Ind_ 376"
"Ind_ 377"	"Ind_ 378"	"Ind_ 379"	"Ind_ 380"
"Ind_ 381"	"Ind_ 382"	"Ind_ 383"	"Ind_ 384"
"Ind_ 385"	"Ind_ 386"	"Ind_ 387"	"Ind_ 388"
"Ind_ 389"	"Ind_ 390"	"Ind_ 391"	"Ind_ 392"
"Ind_ 393"	"Ind_ 394"	"Ind_ 395"	"Ind_ 396"
"Ind_ 397"	"Ind_ 398"	"Ind_ 399"	"Ind_ 400"

Cluster 3 (Cardinal=100) :

"Ind_ 1"	"Ind_ 2"	"Ind_ 3"	"Ind_ 4"
"Ind_ 5"	"Ind_ 6"	"Ind_ 7"	"Ind_ 8"
"Ind_ 9"	"Ind_ 10"	"Ind_ 11"	"Ind_ 12"
"Ind_ 13"	"Ind_ 14"	"Ind_ 15"	"Ind_ 16"
"Ind_ 17"	"Ind_ 18"	"Ind_ 19"	"Ind_ 20"
"Ind_ 21"	"Ind_ 22"	"Ind_ 23"	"Ind_ 24"

"Ind_ 25"	"Ind_ 26"	"Ind_ 27"	"Ind_ 28"
"Ind_ 29"	"Ind_ 30"	"Ind_ 31"	"Ind_ 32"
"Ind_ 33"	"Ind_ 34"	"Ind_ 35"	"Ind_ 36"
"Ind_ 37"	"Ind_ 38"	"Ind_ 39"	"Ind_ 40"
"Ind_ 41"	"Ind_ 42"	"Ind_ 43"	"Ind_ 44"
"Ind_ 45"	"Ind_ 46"	"Ind_ 47"	"Ind_ 48"
"Ind_ 49"	"Ind_ 50"	"Ind_ 51"	"Ind_ 52"
"Ind_ 53"	"Ind_ 54"	"Ind_ 55"	"Ind_ 56"
"Ind_ 57"	"Ind_ 58"	"Ind_ 59"	"Ind_ 60"
"Ind_ 61"	"Ind_ 62"	"Ind_ 63"	"Ind_ 64"
"Ind_ 65"	"Ind_ 66"	"Ind_ 67"	"Ind_ 68"
"Ind_ 69"	"Ind_ 70"	"Ind_ 71"	"Ind_ 72"
"Ind_ 73"	"Ind_ 74"	"Ind_ 75"	"Ind_ 76"
"Ind_ 77"	"Ind_ 78"	"Ind_ 79"	"Ind_ 80"
"Ind_ 81"	"Ind_ 82"	"Ind_ 83"	"Ind_ 84"
"Ind_ 85"	"Ind_ 86"	"Ind_ 87"	"Ind_ 88"
"Ind_ 89"	"Ind_ 90"	"Ind_ 91"	"Ind_ 92"
"Ind_ 93"	"Ind_ 94"	"Ind_ 95"	"Ind_ 96"
"Ind_ 97"	"Ind_ 98"	"Ind_ 99"	"Ind_ 100"

Cluster 4 (Cardinal=100) :

"Ind_ 201"	"Ind_ 202"	"Ind_ 203"	"Ind_ 204"
"Ind_ 205"	"Ind_ 206"	"Ind_ 207"	"Ind_ 208"
"Ind_ 209"	"Ind_ 210"	"Ind_ 211"	"Ind_ 212"
"Ind_ 213"	"Ind_ 214"	"Ind_ 215"	"Ind_ 216"
"Ind_ 217"	"Ind_ 218"	"Ind_ 219"	"Ind_ 220"
"Ind_ 221"	"Ind_ 222"	"Ind_ 223"	"Ind_ 224"
"Ind_ 225"	"Ind_ 226"	"Ind_ 227"	"Ind_ 228"
"Ind_ 229"	"Ind_ 230"	"Ind_ 231"	"Ind_ 232"
"Ind_ 233"	"Ind_ 234"	"Ind_ 235"	"Ind_ 236"
"Ind_ 237"	"Ind_ 238"	"Ind_ 239"	"Ind_ 240"
"Ind_ 241"	"Ind_ 242"	"Ind_ 243"	"Ind_ 244"
"Ind_ 245"	"Ind_ 246"	"Ind_ 247"	"Ind_ 248"
"Ind_ 249"	"Ind_ 250"	"Ind_ 251"	"Ind_ 252"
"Ind_ 253"	"Ind_ 254"	"Ind_ 255"	"Ind_ 256"
"Ind_ 257"	"Ind_ 258"	"Ind_ 259"	"Ind_ 260"
"Ind_ 261"	"Ind_ 262"	"Ind_ 263"	"Ind_ 264"
"Ind_ 265"	"Ind_ 266"	"Ind_ 267"	"Ind_ 268"
"Ind_ 269"	"Ind_ 270"	"Ind_ 271"	"Ind_ 272"
"Ind_ 273"	"Ind_ 274"	"Ind_ 275"	"Ind_ 276"
"Ind_ 277"	"Ind_ 278"	"Ind_ 279"	"Ind_ 280"
"Ind_ 281"	"Ind_ 282"	"Ind_ 283"	"Ind_ 284"
"Ind_ 285"	"Ind_ 286"	"Ind_ 287"	"Ind_ 288"
"Ind_ 289"	"Ind_ 290"	"Ind_ 291"	"Ind_ 292"
"Ind_ 293"	"Ind_ 294"	"Ind_ 295"	"Ind_ 296"
"Ind_ 297"	"Ind_ 298"	"Ind_ 299"	"Ind_ 300"

Inertie expliquée : 73.556140

PARTITION IN 5 CLUSTERS :

Cluster 1 (Cardinal=100) :

"Ind_ 101"	"Ind_ 102"	"Ind_ 103"	"Ind_ 104"
"Ind_ 105"	"Ind_ 106"	"Ind_ 107"	"Ind_ 108"
"Ind_ 109"	"Ind_ 110"	"Ind_ 111"	"Ind_ 112"
"Ind_ 113"	"Ind_ 114"	"Ind_ 115"	"Ind_ 116"
"Ind_ 117"	"Ind_ 118"	"Ind_ 119"	"Ind_ 120"
"Ind_ 121"	"Ind_ 122"	"Ind_ 123"	"Ind_ 124"
"Ind_ 125"	"Ind_ 126"	"Ind_ 127"	"Ind_ 128"
"Ind_ 129"	"Ind_ 130"	"Ind_ 131"	"Ind_ 132"
"Ind_ 133"	"Ind_ 134"	"Ind_ 135"	"Ind_ 136"
"Ind_ 137"	"Ind_ 138"	"Ind_ 139"	"Ind_ 140"
"Ind_ 141"	"Ind_ 142"	"Ind_ 143"	"Ind_ 144"
"Ind_ 145"	"Ind_ 146"	"Ind_ 147"	"Ind_ 148"
"Ind_ 149"	"Ind_ 150"	"Ind_ 151"	"Ind_ 152"
"Ind_ 153"	"Ind_ 154"	"Ind_ 155"	"Ind_ 156"

"Ind_ 157"	"Ind_ 158"	"Ind_ 159"	"Ind_ 160"
"Ind_ 161"	"Ind_ 162"	"Ind_ 163"	"Ind_ 164"
"Ind_ 165"	"Ind_ 166"	"Ind_ 167"	"Ind_ 168"
"Ind_ 169"	"Ind_ 170"	"Ind_ 171"	"Ind_ 172"
"Ind_ 173"	"Ind_ 174"	"Ind_ 175"	"Ind_ 176"
"Ind_ 177"	"Ind_ 178"	"Ind_ 179"	"Ind_ 180"
"Ind_ 181"	"Ind_ 182"	"Ind_ 183"	"Ind_ 184"
"Ind_ 185"	"Ind_ 186"	"Ind_ 187"	"Ind_ 188"
"Ind_ 189"	"Ind_ 190"	"Ind_ 191"	"Ind_ 192"
"Ind_ 193"	"Ind_ 194"	"Ind_ 195"	"Ind_ 196"
"Ind_ 197"	"Ind_ 198"	"Ind_ 199"	"Ind_ 200"

Cluster 2 (Cardinal=52) :

"Ind_ 301"	"Ind_ 302"	"Ind_ 304"	"Ind_ 310"
"Ind_ 312"	"Ind_ 313"	"Ind_ 315"	"Ind_ 317"
"Ind_ 318"	"Ind_ 319"	"Ind_ 320"	"Ind_ 321"
"Ind_ 322"	"Ind_ 325"	"Ind_ 330"	"Ind_ 331"
"Ind_ 332"	"Ind_ 333"	"Ind_ 334"	"Ind_ 336"
"Ind_ 337"	"Ind_ 338"	"Ind_ 339"	"Ind_ 341"
"Ind_ 345"	"Ind_ 349"	"Ind_ 351"	"Ind_ 352"
"Ind_ 354"	"Ind_ 356"	"Ind_ 358"	"Ind_ 359"
"Ind_ 360"	"Ind_ 361"	"Ind_ 362"	"Ind_ 363"
"Ind_ 366"	"Ind_ 367"	"Ind_ 369"	"Ind_ 373"
"Ind_ 374"	"Ind_ 375"	"Ind_ 376"	"Ind_ 378"
"Ind_ 379"	"Ind_ 382"	"Ind_ 385"	"Ind_ 386"
"Ind_ 389"	"Ind_ 390"	"Ind_ 396"	"Ind_ 399"

Cluster 3 (Cardinal=100) :

"Ind_ 1"	"Ind_ 2"	"Ind_ 3"	"Ind_ 4"
"Ind_ 5"	"Ind_ 6"	"Ind_ 7"	"Ind_ 8"
"Ind_ 9"	"Ind_ 10"	"Ind_ 11"	"Ind_ 12"
"Ind_ 13"	"Ind_ 14"	"Ind_ 15"	"Ind_ 16"
"Ind_ 17"	"Ind_ 18"	"Ind_ 19"	"Ind_ 20"
"Ind_ 21"	"Ind_ 22"	"Ind_ 23"	"Ind_ 24"
"Ind_ 25"	"Ind_ 26"	"Ind_ 27"	"Ind_ 28"
"Ind_ 29"	"Ind_ 30"	"Ind_ 31"	"Ind_ 32"
"Ind_ 33"	"Ind_ 34"	"Ind_ 35"	"Ind_ 36"
"Ind_ 37"	"Ind_ 38"	"Ind_ 39"	"Ind_ 40"
"Ind_ 41"	"Ind_ 42"	"Ind_ 43"	"Ind_ 44"
"Ind_ 45"	"Ind_ 46"	"Ind_ 47"	"Ind_ 48"
"Ind_ 49"	"Ind_ 50"	"Ind_ 51"	"Ind_ 52"
"Ind_ 53"	"Ind_ 54"	"Ind_ 55"	"Ind_ 56"
"Ind_ 57"	"Ind_ 58"	"Ind_ 59"	"Ind_ 60"
"Ind_ 61"	"Ind_ 62"	"Ind_ 63"	"Ind_ 64"
"Ind_ 65"	"Ind_ 66"	"Ind_ 67"	"Ind_ 68"
"Ind_ 69"	"Ind_ 70"	"Ind_ 71"	"Ind_ 72"
"Ind_ 73"	"Ind_ 74"	"Ind_ 75"	"Ind_ 76"
"Ind_ 77"	"Ind_ 78"	"Ind_ 79"	"Ind_ 80"
"Ind_ 81"	"Ind_ 82"	"Ind_ 83"	"Ind_ 84"
"Ind_ 85"	"Ind_ 86"	"Ind_ 87"	"Ind_ 88"
"Ind_ 89"	"Ind_ 90"	"Ind_ 91"	"Ind_ 92"
"Ind_ 93"	"Ind_ 94"	"Ind_ 95"	"Ind_ 96"
"Ind_ 97"	"Ind_ 98"	"Ind_ 99"	"Ind_ 100"

Cluster 4 (Cardinal=48) :

"Ind_ 303"	"Ind_ 305"	"Ind_ 306"	"Ind_ 307"
"Ind_ 308"	"Ind_ 309"	"Ind_ 311"	"Ind_ 314"
"Ind_ 316"	"Ind_ 323"	"Ind_ 324"	"Ind_ 326"
"Ind_ 327"	"Ind_ 328"	"Ind_ 329"	"Ind_ 335"
"Ind_ 340"	"Ind_ 342"	"Ind_ 343"	"Ind_ 344"
"Ind_ 346"	"Ind_ 347"	"Ind_ 348"	"Ind_ 350"
"Ind_ 353"	"Ind_ 355"	"Ind_ 357"	"Ind_ 364"
"Ind_ 365"	"Ind_ 368"	"Ind_ 370"	"Ind_ 371"
"Ind_ 372"	"Ind_ 377"	"Ind_ 380"	"Ind_ 381"
"Ind_ 383"	"Ind_ 384"	"Ind_ 387"	"Ind_ 388"
"Ind_ 391"	"Ind_ 392"	"Ind_ 393"	"Ind_ 394"
"Ind_ 395"	"Ind_ 397"	"Ind_ 398"	"Ind_ 400"

```

Cluster 5 (Cardinal=100) :
"Ind_ 201"      "Ind_ 202"      "Ind_ 203"      "Ind_ 204"
"Ind_ 205"      "Ind_ 206"      "Ind_ 207"      "Ind_ 208"
"Ind_ 209"      "Ind_ 210"      "Ind_ 211"      "Ind_ 212"
"Ind_ 213"      "Ind_ 214"      "Ind_ 215"      "Ind_ 216"
"Ind_ 217"      "Ind_ 218"      "Ind_ 219"      "Ind_ 220"
"Ind_ 221"      "Ind_ 222"      "Ind_ 223"      "Ind_ 224"
"Ind_ 225"      "Ind_ 226"      "Ind_ 227"      "Ind_ 228"
"Ind_ 229"      "Ind_ 230"      "Ind_ 231"      "Ind_ 232"
"Ind_ 233"      "Ind_ 234"      "Ind_ 235"      "Ind_ 236"
"Ind_ 237"      "Ind_ 238"      "Ind_ 239"      "Ind_ 240"
"Ind_ 241"      "Ind_ 242"      "Ind_ 243"      "Ind_ 244"
"Ind_ 245"      "Ind_ 246"      "Ind_ 247"      "Ind_ 248"
"Ind_ 249"      "Ind_ 250"      "Ind_ 251"      "Ind_ 252"
"Ind_ 253"      "Ind_ 254"      "Ind_ 255"      "Ind_ 256"
"Ind_ 257"      "Ind_ 258"      "Ind_ 259"      "Ind_ 260"
"Ind_ 261"      "Ind_ 262"      "Ind_ 263"      "Ind_ 264"
"Ind_ 265"      "Ind_ 266"      "Ind_ 267"      "Ind_ 268"
"Ind_ 269"      "Ind_ 270"      "Ind_ 271"      "Ind_ 272"
"Ind_ 273"      "Ind_ 274"      "Ind_ 275"      "Ind_ 276"
"Ind_ 277"      "Ind_ 278"      "Ind_ 279"      "Ind_ 280"
"Ind_ 281"      "Ind_ 282"      "Ind_ 283"      "Ind_ 284"
"Ind_ 285"      "Ind_ 286"      "Ind_ 287"      "Ind_ 288"
"Ind_ 289"      "Ind_ 290"      "Ind_ 291"      "Ind_ 292"
"Ind_ 293"      "Ind_ 294"      "Ind_ 295"      "Ind_ 296"
"Ind_ 297"      "Ind_ 298"      "Ind_ 299"      "Ind_ 300"

```

Inertie expliquée : 74.935952

Si nous faisons un graphique de l'évolution de l'inertie expliquée, nous obtenons :

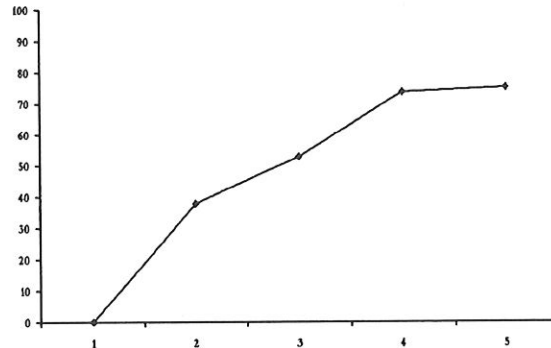


FIGURE 14. Evolution de l'inertie expliquée en fonction du nombre de classes

Le nombre de classes est bel et bien de 4 car l'inertie expliquée par une classification en 5 classes est minime.

Nous regroupons également ces résultats dans un fichier de sortie : `scluster.txt`.

11.2.3. Comparaison des méthodes

Comme pour le jeu de données réelles, l'évolution de l'inertie expliquée en fonction du nombre de classes pour les deux méthodes est semblable.

	DIV	DYCLASS
2 classes	37.79	37.07
3 classes	56.28	52.74
4 classes	74.06	73.56
5 classes	75.05	74.94

Table 1: Inerties expliquées selon les méthodes

On peut prouver que l'inertie expliquée est relativement faible car dans le jeu de données précis que nous avons créé, les hypercubes-enveloppes des classes sont contigus. L'inertie inter-classe est donc encore assez élevée. Si par exemple, nous initialisons les "milieux moyens" à 2 et la longueur à 1 sans changer les autres valeurs, nous obtenons :

	DIV	DYCLASS
2 classes	46.25	46.15
3 classes	69.09	69.00
4 classes	91.51	91.26
5 classes	91.83	91.59

Table 2: Inerties expliquées selon les méthodes

Ensuite, si nous appliquons le programme de comparaison (`comparaison.cpp`) aux fichiers de données créés précédemment (`chavent.txt` et `scluster.txt`), on crée le fichier de résultats suivant : `resultat.txt`.

Matrice de contingence pour 2 classes :

```
200    0
0      200
```

```
Valeur de chi2 pour 2 classes : 400.000000
Valeur de max pour 2 classes : 400 / 400 => Pourcentage de bien classes : 100
```

```
Matrice de contingence pour 3 classes :
```

```
0      0      100
177    0      0
23     100    0
```

```
Valeur de chi2 pour 3 classes : 766.805000
Valeur de max pour 3 classes : 377 / 400 => Pourcentage de bien classes : 94
```

```
Matrice de contingence pour 4 classes :
```

```
0      0      0      100
100    0      0      0
0      0      100    0
0      100    0      0
```

```
Valeur de chi2 pour 4 classes : 1200.000000
Valeur de max pour 4 classes : 400 / 400 => Pourcentage de bien classes : 100
```

```
Matrice de contingence pour 5 classes :
```

```
0      0      0      100    0
45     0      0      0      7
0      0      100    0      0
10     0      0      0      38
0      100    0      0      0
```

```
Valeur de chi2 pour 5 classes : 1701.125000
Valeur de max pour 5 classes : 383 / 400 => Pourcentage de bien classes : 95
```

Les valeurs obtenues pour le test du χ^2 montrent, comme pour le jeu de données précédent, qu'il n'y a pas d'indépendance, les deux méthodes de classification sont bien plus ou moins semblables. Les résultats pour la maximisation "ligne-colonne" montrent que les partitions en 4 classes sont identiques. Les deux méthodes retrouvent donc bien les classes convexes que nous avons artificiellement créées.

CONCLUSION

L'étude théorique et les applications nous permettent de donner quelques avantages et inconvénients pour les deux méthodes.

En ce qui concerne la méthode divisive hiérarchique de Chavent, un avantage certain est le fait que les partitions successives dépendent les unes des autres : une partition en $k + 1$ classes n'étant que la partition en k classes dont une classe a été subdivisée. Cependant, cette subdivision est également un inconvénient car elle ne donne pas la possibilité de remettre en question les "partitions-mères" de la hiérarchie : lorsqu'une "voie" de classement a été choisie, il n'y a pas moyen de revenir en arrière.

En ce qui concerne la méthode des nuées dynamiques de Diday et Lechevallier, le problème évoqué plus haut ne se pose pas. Le seul gros inconvénient pourrait être le fait que pour initialiser cette méthode, il nous faut savoir le nombre de classes que nous désirons à l'avance. Pour une étude comme celle-ci où nous ne nous sommes pas préoccupés de la détermination du nombre de classes, nous sommes obligés d'appliquer $k - 1$ fois la méthode si nous voulons connaître les différentes partitions jusqu'à k classes.

L'avantage global du travail est cependant le fait d'avoir utilisé des données symboliques intervalles et d'avoir surmonté les problèmes de définition de distance ou d'inertie pour ce type de données.

Des perspectives intéressantes seraient donc, par exemple :

- ◇ une remise en question éventuelle des choix binaires dans la méthode divisive de Chavent ainsi que l'implémentation de cette remise en question : possibilité envisagée par Chavent elle-même [6].
- ◇ un lien entre la détermination du nombre de classes et les méthodes vues.
- ◇ une comparaison avec d'autres méthodes de classification suffisamment semblables pour pouvoir en tirer des informations (SCLASS,...).
- ◇ une utilisation d'autres données symboliques : les données (multi-)modales, (multi-)nominales voire hiérarchiques ou en histogrammes.
- ◇ une amélioration des indices expliquant la "qualité" d'une partition.
- ◇ une amélioration et de nouveaux indices pour comparer deux ou plusieurs partitions d'un même ensemble d'individus.

Cependant, il faut également rappeler que les méthodes de classification ne font jamais de miracles. Si nous savons toujours classer les individus (par maximisation ou minimisation d'un critère mathématique), il faut toujours bien distinguer ce problème purement mathématique du problème de la "qualité intuitive" d'une solution (existence réelle de classes naturelles,...).

ANNEXES

Annexe A

Extension de la mesure de l'inertie d'une nuage de point en le calculant indépendamment de son centre de gravité, c'est-à-dire, en ne faisant intervenir que les distance euclidiennes entre individus.

Cette proposition est utilisée dans le cadre de l'inertie d'une classe C_k (section 3.3.1) et de l'extension du critère d'inertie (section 7.2) :

$$W_k = \sum_{x_i \in C_k} p_i d_M^2(x_i, g_k) .$$

Proposition :

Soit Ω un ensemble de n individus décrits dans \mathbb{R}^p par p vecteurs x_i munis des poids p_i ($i = 1, \dots, n$).

Soient :

- ◇ $C = (C_1, \dots, C_m)$ une partition en m classes de Ω ,
- ◇ M une matrice $p \times p$ symétrique définie positive,
- ◇ μ_k le poids de la classe C_k : $\mu_k = \sum_{x_i \in C_k} p_i$,
- ◇ g_k est le centre de gravité de la classe C_k : $g_k = \frac{1}{\mu_k} \sum_{x_i \in C_k} p_i x_i$,

◇ $d_M^2(x_i, x_{i'}) = (x_i - x_{i'})' M (x_i - x_{i'})$. Par exemple, si $M = I$, d_M est la distance euclidienne simple, si $M = \Sigma^{-1}$ (l'inverse de la matrice variance-covariance des individus) d_M est la distance de Mahalanobis.

Alors l'inertie associée à la classe C_k est :

$$W_k = \sum_{x_i \in C_k} p_i d_M^2(x_i, g_k) = \sum_{x_i \in C_k} \sum_{x_{i'} \in C_k} \frac{p_i p_{i'}}{2\mu_k} d_M^2(x_i, x_{i'})$$

Démonstration :

Nous avons besoin de la définition de l'inertie par rapport à un point et du théorème de Huygens.

Soit g_k le centre de gravité d'une classe C_k et $x_i \in C_k$. Par le théorème de Huygens, l'inertie du nuage $N(C_k)$ par rapport à x_i s'écrit :

$$I_{x_i} = I_{g_k} + \underbrace{\sum_{x_i \in C_k} p_i}_{\mu_k} d_M^2(x_i, g_k)$$

Multiplions l'équation par p_i et sommions pour tous les $x_i \in C_k$:

$$\sum_{x_i \in C_k} p_i I_{x_i} = \underbrace{\sum_{x_i \in C_k} p_i}_{\mu_k} I_{g_k} + \underbrace{\mu_k \sum_{x_i \in C_k} p_i d_M^2(x_i, g_k)}_{I_{g_k}}$$

Or nous savons que :

$$I_{x_i} = \sum_{x_{i'} \in C_k} p_{i'} d_M^2(x_i, x_{i'})$$

En remplaçant dans l'expression obtenue auparavant :

$$\sum_{x_i \in C_k} \sum_{x_{i'} \in C_k} p_i p_{i'} d_M^2(x_i, x_{i'}) = \mu_k I_{g_k} + \mu_k I_{g_k}$$

Nous retrouvons bien le résultat :

$$W_k = I_{g_k} = \sum_{x_i \in C_k} \sum_{x_{i'} \in C_k} \frac{p_i p_{i'}}{2\mu_k} d_M^2(x_i, x_{i'})$$

□

Annexe B : Programmes

Programme "comparaison.cpp"

```

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <iostream.h>
#include <math.h>

FILE *file;
FILE *res;

void main()
{
    void lecture (char *namefile, int **classe, int m);
    int i,j,k,l,m,n;
    int **scluster;
    int **chavent;
    char stringbuf[2000];

    file=fopen("scluster.txt","r");
    fgets(stringbuf,2000,file);
    n=atoi(stringbuf);
    fgets(stringbuf,2000,file);
    m=atoi(stringbuf);
    fclose(file);

    //      création des deux tableaux dynamiques (individus*classes) en vue de la comparaison

    scluster = new int *[n];
    chavent = new int *[n];
    for (i=0;i<n;i++)
    {
        scluster[i] = new int [m-1];
        chavent[i] = new int [m-1];
    }

    //      lecture des fichiers de données et remplissage des tableaux dynamiques

    lecture("scluster.txt", scluster, m-1);
    lecture("chavent.txt", chavent, m-1);

    //      ouverture du fichier de résultats

    res=fopen("resultat.txt","w+");

    //      création de la matrice de contingence

    int **resultat;
    resultat = new int *[m];
    for (i=0;i<m;i++)
        resultat[i]= new int[m];

    //      boucle sur les matrices de contingence successives (2 classes -> ...)

    for (k=0;k<m-1;k++)
    {
        for (i=0;i<k+2;i++)
            for (j=0;j<k+2;j++)
                resultat[i][j]=0;

    }

    //      remplissage de la matrice de contingence par importation des données des tableaux des méthodes

    for (i=0;i<n;i++)
        resultat[(scluster[i][k])-1][(chavent[i][k])-1]++;

```

```

// écriture de la matrice de contingence dans le fichier de résultats

fprintf(res,"Matrice de contingence pour %i classes :\n\n",k+2);
for (i=0;i<k+2;i++)
{
    for (j=0;j<k+2;j++)
        fprintf(res,"%i\t",resultat[i][j]);
    fprintf(res,"\n");
}
fprintf(res,"\n");

// implémentation du test du chi2

double chi2=0;
for (i=0;i<k+2;i++)
{
    for (j=0;j<k+2;j++)
    {
        chi2=chi2+(pow((resultat[i][j]-(double(n)/pow(k+2,2))),2))/
            (double(n)/pow(k+2,2));
    }
}
fprintf(res,"Valeur de chi2 pour %2i classes : %f \n",k+2,chi2);

// implémentation de la méthode de la maximisation d'éléments communs

int max=0,hor,vert;
for (l=0;l<k+2;l++)
{
    int maxtemp=0;

// on recherche la valeur maximale du tableau de contingence

    for (i=0;i<k+2;i++)
    {
        for (j=0;j<k+2;j++)
        {
            if (maxtemp<resultat[i][j])
            {
                maxtemp=resultat[i][j];
                hor=i;
                vert=j;
            }
        }
    }

// on la garde en mémoire et on annule toutes les valeurs de la ligne et de la
// colonne qui lui correspond

    max=max+maxtemp;
    for (i=0;i<k+2;i++)
        resultat[i][vert]=0;
    for (j=0;j<k+2;j++)
        resultat[hor][j]=0;
}

// calcul du pourcentage d'individus communs par rapport au nombre d'individus total

float procent;
procent=100*(float(max)/float(n));
fprintf(res,"Valeur de max pour %2i classes : %i / %i ",k+2,max,n);
fprintf(res,"=> Pourcentage de bien classes : %i %\n\n",int(procent));
}

// désallocation de la mémoire

```

```

    for (i=0;i<n;i++)
    {
        delete[] scluster[i];
        delete[] chavent[i];
    }
    delete[] scluster;
    delete[] chavent;

    for (i=0;i<m;i++)
        delete[] resultat[i];
    delete[] resultat;
}

void lecture(char *namefile, int **classe, int m)
{
    char stringbuf[2000], var[256], classement[256];
    char *position;
    int result, i=0, j=0;

    file=fopen(namefile,"r");
    fgets(stringbuf,2000,file);
    fgets(stringbuf,2000,file);
    while (fgetc(file) != -1)
    {
        // lecture d'une ligne du fichier de données et séparation du nom de la variable et de ses
        // classements

        fgets(stringbuf,2000,file);
        position=strchr(stringbuf,'\t');
        result = position - stringbuf;
        strncpy(var,stringbuf,result);
        strncpy(classement,position+1,strlen(stringbuf)-result);
        var[result]='\0';
        classement[strlen(stringbuf)-result]='\0';

        // découpage du classement et remplissage du tableau du tableau dynamique

        for (j=0;j<m;j++)
        {
            classe[i][j]=atoi(classement);
            position=strchr(classement,'\t');
            result = position - classement;
            strncpy(classement,position+1,strlen(classement)-result);
            classement[strlen(stringbuf)-result]='\0';
        }
        i++;
    }
    fclose(file);
}

```

Programme "données.cpp"

```

#include "stdafx.h"
#include <stdlib.h>
#include <stdio.h>
#include <fstream.h>
#include <time.h>
#include <math.h>

void main()
{
    int m,n,p;
    int i,j,k,l;
    double centre,longueur;

```



```

//      requête d'éléments

cout << "Dimension de l'espace de representation : "; cin >> p;
cout << "Nombre de classes artificiellement creees : "; cin >> m;
cout << "Nombre d'intervalles dans chaque classe : "; cin >> n;

//      création d'un tableau dynamiques à 3 dimensions (individus*variables*4)

double ***ind;
ind = new double **[n];
for (i=0;i<n;i++)
{
    ind[i] = new double *[p];
    for(j=0;j<p;j++)
        ind[i][j] = new double [4];
}

//      création d'un tableau dynamique de nombres aléatoires

double ***random;
random = new double **[n];
for (i=0;i<n;i++)
{
    random[i] = new double *[p];
    for(j=0;j<p;j++)
        random[i][j] = new double [2];
}

//      création d'un vecteur dynamique dont la taille est le nombre de variables

double *vector;
vector = new double[p];

//      ouverture du fichier de sortie

FILE *lis;
lis=fopen("liste.txt","w+");

//      test sur la possibilité de créer les classes par rapport au nombre de variables

if (m<pow(2,p))
{

//      initialisation du vecteur selon les "milieux moyens" par variables

    cout << "Coordonnees des \"milieux moyens\" par variables : "; cin >> centre;
    for (i=0;i<p;i++)
        vector[i]=centre;

    fprintf(lis,"Classes\tIndividus");
    for (i=0;i<p;i++)
        fprintf(lis,"\tVar%i",i+1);
    fprintf(lis,"\n");

    cout << "Longueur maximale des intervalles : "; cin >> longueur;

//      boucle sur le nombre de classes

    for (l=0;l<m;l++)
    {

//      calcul de nombres aléatoires et remplissage du tableau

        srand(1);
        for(i=0;i<n;i++)
        {

```

```

        for(j=0;j<p;j++)
        {
            for(k=0;k<2;k++)
                random[i][j][k]=longueur*(double(rand())/double(32767));
        }
    }

// calcul des coordonnées "m-1" et "min-max" des intervalles
    for(i=0;i<n;i++)
    {
        for (j=0;j<p;j++)
        {
            ind[i][j][0]=vector[j]+random[i][j][0]-(longueur/2);
            ind[i][j][1]=random[i][j][1];
            ind[i][j][2]=ind[i][j][0]-ind[i][j][1]/2;
            ind[i][j][3]=ind[i][j][0]+ind[i][j][1]/2;
        }
    }

// écriture dans le fichier de sortie des intervalles obtenus
    for (i=0;i<n;i++)
    {
        fprintf(lis,"Cla_%i\tInd_%4i",1+1,(1*n)+i+1);
        for (j=0;j<p;j++)
            fprintf(lis,"\t%f",ind[i][j][2]);
        fprintf(lis,"\n");
        fprintf(lis,"Cla_%i\tInd_%4i",1+1,(1*n)+i+1);
        for (j=0;j<p;j++)
            fprintf(lis,"\t%f",ind[i][j][3]);
        fprintf(lis,"\n");
    }

// modification du vecteur des "milieux moyens" par variables
    if (vector[p-1] == centre)
        vector[p-1]=-centre;
    else
    {
        i=p-1;
        while (vector[i] != centre)
            i--;
        vector[i]=-centre;
        for (j=i+1;j<p;j++)
            vector[j]=centre;
    }
}
else
{
    printf("Impossible de creer les donnees artificielles : nombre de classes");
    printf(" trop eleve\npar rapport a la dimension de l'espace de representation\n");
}

// désallocation de la mémoire
for (i=1;i<n;i++)
{
    for (j=1;j<p;j++)
    {
        delete[] random[i][j];
        delete[] ind[i][j];
    }
}
delete[] random;
delete[] ind;

```

```

        delete[] vector;
    }

```

Annexe C : Données réelles

Fichier "temp1988.sds"

```

SODAS = (
CONTAINS = (
    FILES, HEADER, INDIVIDUALS, VARIABLES, RECTANGLE_MATRIX
),
FILE = (
    procedure_name = "db2so" ,
    version = "sans" ,
    create_date = ""
),
HEADER = (
    title = "la description de l'année 1988" ,
    sub_title = "température" ,
    indiv_nb = 60 ,
    var_nb = 13 ,
    rules_nb = 0 ,
    nb_var_set = 0 ,
    nb_indiv_set = 0 ,
    nb_var_nom = 0 ,
    nb_var_cont = 0 ,
    nb_var_text = 0 ,
    nb_var_cont_symb = 12 ,
    nb_var_nom_symb = 1 ,
    nb_var_nom_mod = 0 ,
    nb_na = 0 ,
    nb_null = 0 ,
    nb_nu = 0 ,
    nb_hierarchies = 0
),
INDIVIDUALS = (
    (0,"AA00", "AnQing_1988" ),
    (1,"AA01", "BaoDing_1988" ),
    (2,"AA02", "BeiJing_1988" ),
    (3,"AA03", "BoKeTu_1988" ),
    (4,"AA04", "ChangChun_1988" ),
    (5,"AA05", "ChangSha_1988" ),
    (6,"AA06", "ChengDu_1988" ),
    (7,"AA07", "ChongQing_1988" ),
    (8,"AA08", "DaLian_1988" ),
    (9,"AA09", "FuZhou_1988" ),
    (10,"AA10", "GuangZhou_1988" ),
    (11,"AA11", "GuiYang_1988" ),
    (12,"AA12", "HaErBin_1988" ),
    (13,"AA13", "HaiKou_1988" ),
    (14,"AA14", "Hailaer_1988" ),
    (15,"AA15", "HaMi_1988" ),
    (16,"AA16", "HangZhou_1988" ),
    (17,"AA17", "HanZhong_1988" ),
    (18,"AA18", "HuHeHaoTe_1988" ),
    (19,"AA19", "JiNan_1988" ),
    (20,"AA20", "Jiu Quan_1988" ),
    (21,"AA21", "KunMing_1988" ),
    (22,"AA22", "LanZhou_1988" ),
    (23,"AA23", "LaSa_1988" ),
    (24,"AA24", "LiuZhou_1988" ),
    (25,"AA25", "MuDanJiang_1988" ),
    (26,"AA26", "NanChang_1988" ),
    (27,"AA27", "NanJing_1988" ),

```

```

(28,"AA28", "NanNing_1988" ),
(29,"AA29", "NenJiang_1988" ),
(30,"AA30", "QingDao_1988" ),
(31,"AA31", "QingJiang_1988" ),
(32,"AA32", "QiQiHaEr_1988" ),
(33,"AA33", "QuZhou_1988" ),
(34,"AA34", "ShangHai_1988" ),
(35,"AA35", "ShanTou_1988" ),
(36,"AA36", "ShenYang_1988" ),
(37,"AA37", "TaiYuan_1988" ),
(38,"AA38", "TengChong_1988" ),
(39,"AA39", "TianJin_1988" ),
(40,"AA40", "TianShui_1988" ),
(41,"AA41", "WenZhou_1988" ),
(42,"AA42", "WuHan_1988" ),
(43,"AA43", "WuLuMuQi_1988" ),
(44,"AA44", "WuZhou_1988" ),
(45,"AA45", "XiaMen_1988" ),
(46,"AA46", "XiAn_1988" ),
(47,"AA47", "XiChang_1988" ),
(48,"AA48", "XiNing_1988" ),
(49,"AA49", "XuZhou_1988" ),
(50,"AA50", "YanTai_1988" ),
(51,"AA51", "Yi Ning_1988" ),
(52,"AA52", "YiChang_1988" ),
(53,"AA53", "YinChuan_1988" ),
(54,"AA54", "YongAn_1988" ),
(55,"AA55", "YuLin_1988" ),
(56,"AA56", "ZhangYe_1988" ),
(57,"AA57", "ZhanJiang_1988" ),
(58,"AA58", "ZhengZhou_1988" ),
(59,"AA59", "ZhiJiang_1988" )
),
VARIABLES = (
(1 ,inter_cont ,"" , "AB00" , "January" ,0, 0, -28.4, 23),
(2 ,inter_cont ,"" , "AC00" , "February" ,0, 0, -29.6, 23),
(3 ,inter_cont ,"" , "AD00" , "March" ,0, 0, -20.2, 24.9),
(4 ,inter_cont ,"" , "AE00" , "April" ,0, 0, -4.5, 26.9),
(5 ,inter_cont ,"" , "AF00" , "May" ,0, 0, 1.9, 33.4),
(6 ,inter_cont ,"" , "AG00" , "June" ,0, 0, 9.3, 34.3),
(7 ,inter_cont ,"" , "AH00" , "July" ,0, 0, 10.8, 36.9),
(8 ,inter_cont ,"" , "AI00" , "August" ,0, 0, 10.2, 34.8),
(9 ,inter_cont ,"" , "AJ00" , "September" ,0, 0, 4.1, 31.9),
(10 ,inter_cont ,"" , "AK00" , "October" ,0, 0, -4, 28.5),
(11 ,inter_cont ,"" , "AL00" , "November" ,0, 0, -14.4, 24.3),
(12 ,inter_cont ,"" , "AM00" , "December" ,0, 0, -26.1, 22.4),
(13 ,mult_nominal ,"" , "AN00" , "Humidity" ,0, 0, 12, (
(1 , "AN01" , "January" ,0),
(2 , "AN02" , "June" ,0),
(3 , "AN03" , "February" ,0),
(4 , "AN04" , "September" ,0),
(5 , "AN05" , "May" ,0),
(6 , "AN06" , "April" ,0),
(7 , "AN07" , "July" ,0),
(8 , "AN08" , "March" ,0),
(9 , "AN09" , "August" ,0),
(10 , "AN10" , "November" ,0),
(11 , "AN11" , "December" ,0),
(12 , "AN12" , "October" ,0 ) )
)
),
RECTANGLE_MATRIX = (
(( 1.8 : 7.1 ), ( 2.1 : 7.2 ), ( 5.2 : 11.2 ), ( 13.5 : 21.9 ), ( 18.6 : 25.8 ), ( 22.9 : 29.5 ),
( 27.5 : 35.3 ), ( 24.3 : 31.2 ), ( 20.1 : 26.2 ), ( 15.5 : 22.3 ), ( 7.8 : 17.9 ),
( 4.3 : 11.8 ), (1,2,3,4,5,6,7,8,9,10,11,12)),
(( -7.1 : 1.7 ), ( -5.3 : 4.8 ), ( 0.2 : 10.8 ), ( 9.1 : 22.2 ), ( 14.4 : 26.1 ),
( 20 : 31.7 ), ( 22.4 : 30.5 ), ( 21.6 : 28.8 ), ( 16.7 : 26.6 ), ( 9.5 : 20.5 ),

```

(0.8 : 14), (-3.9 : 5.2), (1,2,4,5,7,9,11,12)),
 ((-7.2 : 2.1), (-5.9 : 3.8), (-0.6 : 9.6), (8.7 : 21.1), (14.4 : 26), (19.7 : 30.5),
 (22.1 : 30.5), (21.2 : 28.6), (16.6 : 26.7), (8.7 : 20.4), (1.5 : 12.7),
 (-4.4 : 4.7), (2,4,7,9,12)),
 ((-23.4 : -15.5), (-24 : -14), (-17.6 : -4.7), (-4.5 : 9.5), (1.9 : 15.2),
 (9.3 : 23), (12.9 : 23), (10.8 : 23.6), (4.1 : 17), (-4 : 8.9), (-13.5 : -4.2),
 (-21.1 : -13.1), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-16.9 : -6.7), (-17.6 : -6.8), (-9.3 : 2.7), (1.1 : 12.7), (8.6 : 20.4),
 (16.2 : 27.5), (19.8 : 28), (18.7 : 26.5), (11 : 21.3), (2.6 : 14.2),
 (-7.9 : 2.2), (-15.9 : -7.2), (1,2,4,5,6,7,9,10,11,12)),
 ((2.7 : 7.4), (3.1 : 7.7), (6.5 : 12.6), (12.9 : 22.9), (19.2 : 26.8), (21.9 : 31),
 (25.7 : 34.8), (24.4 : 32), (20 : 27), (15.3 : 22.8), (7.6 : 19.6),
 (4.1 : 13.3), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((2.2 : 10.5), (3.4 : 9.9), (5.5 : 11.9), (11.3 : 21.8), (17.4 : 26.5),
 (21.4 : 29.3), (21.9 : 28.9), (21.1 : 27.4), (18.1 : 24.2), (14.3 : 19.4),
 (9.8 : 16.7), (5.3 : 10.8), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((6.6 : 10.9), (6.9 : 11.3), (8.8 : 13.8), (14.1 : 23.3), (19.7 : 27.7),
 (22.4 : 30.7), (25 : 33.8), (24.1 : 32.2), (19.8 : 25.2), (15.8 : 20.5),
 (12.2 : 17.1), (8.9 : 12.3), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-4.7 : 1.3), (-5.9 : 0.6), (-0.8 : 6), (6.3 : 14.4), (12.5 : 20.3), (17.3 : 24.6),
 (20.9 : 26), (21.9 : 27.5), (18.1 : 24.1), (12.3 : 18.8), (3 : 9.8),
 (-2.8 : 3.4), (1,2,4,5,7,9,11,12)),
 ((9.8 : 16.6), (8 : 13.8), (9.2 : 15.2), (13.6 : 22.4), (19.5 : 25.9), (23.7 : 32.5),
 (26.2 : 35.9), (25 : 34.8), (22.4 : 28.6), (18.7 : 25.1), (12.4 : 20.7),
 (8.8 : 18.6), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((12.2 : 20.6), (11.4 : 17.7), (12.7 : 17.9), (17.4 : 24.3), (24.2 : 30.3),
 (25.1 : 32.8), (25.3 : 33.6), (25 : 31.9), (24.1 : 31.4), (21 : 28.5),
 (14.2 : 23.4), (10.9 : 22.2), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((3.3 : 9.9), (2.1 : 7.8), (5.1 : 12.9), (11.1 : 20.8), (17.1 : 24.8), (18.7 : 27.6),
 (21.4 : 29.2), (19.5 : 27.6), (16.6 : 24), (13.2 : 19.9), (9 : 17.4),
 (6.7 : 13.8), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-20 : -9.6), (-22.3 : -10), (-10.8 : 1.7), (0.2 : 11.9), (7.5 : 19), (14.6 : 27.3),
 (17.8 : 27.2), (17.7 : 26.3), (8.9 : 20.8), (-0.2 : 12.5), (-9.6 : 0.4),
 (-21.6 : -10.1), (1,2,3,4,5,6,7,9,10,11,12)),
 ((17.3 : 23), (17.3 : 23), (17.1 : 23.7), (19.9 : 25.9), (25.3 : 33.4),
 (25.5 : 33.2), (25.8 : 34), (25 : 32.6), (25.1 : 31.5), (23 : 26.7),
 (18.6 : 22.6), (16.3 : 21.6), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-28.4 : -19.1), (-29.6 : -18.1), (-20.2 : -7.9), (-2.9 : 10), (3.8 : 16.5),
 (12.5 : 24.2), (14.7 : 25.4), (13.7 : 24.8), (5.3 : 17.6), (-3.2 : 9.8),
 (-13.8 : -3.7), (-26 : -17.2), (1,2,3,4,6,7,8,9,10,11,12)),
 ((-13.8 : -2.4), (-12.8 : -1), (-3.6 : 10), (5.2 : 20.8), (11.7 : 25.7),
 (17 : 32), (17.9 : 34), (17.6 : 33.7), (13.2 : 29.6), (3.2 : 18.1),
 (-4.4 : 8.9), (-9.9 : 1.4), (1,3,11,12)),
 ((2.5 : 8.7), (1.9 : 7.3), (4.7 : 11.6), (11.8 : 21.2), (16.9 : 25.1), (21.7 : 29.5),
 (26.1 : 35.8), (23.5 : 30.8), (19.9 : 26.6), (15.2 : 23), (6.9 : 18),
 (3.4 : 12.6), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-0.4 : 8.1), (0.7 : 7.2), (3.3 : 10), (9.9 : 21.1), (16.4 : 26.5), (20.2 : 30.1),
 (22.4 : 29.4), (21 : 28.3), (16.9 : 23.8), (12.2 : 17.7), (4.6 : 14.9),
 (0.7 : 8.3), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-16.5 : -6.1), (-13.6 : -1.5), (-7.8 : 3.7), (0.4 : 15.1), (8.9 : 22.5),
 (14.4 : 27.6), (17.7 : 28), (15.6 : 24.4), (8.9 : 22.1), (1.6 : 14.6),
 (-7.4 : 5.3), (-12.8 : -2.5), (1,4,7,9,11,12)),
 ((-2.6 : 4.5), (-2 : 5.7), (2.7 : 11.5), (11.9 : 21.9), (16.6 : 26.7), (22.1 : 33.2),
 (22.6 : 30.7), (22.1 : 30.4), (17.9 : 27.8), (11.9 : 22), (4.8 : 15.1),
 (-0.9 : 6.2), (2,4,5,7,9,11,12)),
 ((-14 : -0.9), (-12.1 : -1.2), (-6.2 : 5.7), (2.8 : 17.6), (8.3 : 21.6),
 (12.5 : 26.1), (15 : 29.4), (13.9 : 28), (9.1 : 22.9), (1 : 14.1),
 (-5.2 : 8.1), (-11.3 : 1.2), (2,3,4,9,11,12)),
 ((2.8 : 16.6), (4 : 19.4), (6.7 : 21.4), (10.4 : 23.4), (15.9 : 25.3), (16.4 : 25.3),
 (16.9 : 24.4), (16.6 : 23.8), (14 : 21.7), (12.4 : 19.7), (7.4 : 16.3),
 (3.8 : 15.5), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-9.4 : 2.6), (-6.5 : 3.7), (-1.2 : 8.8), (6.1 : 19.6), (11 : 23.4), (14.3 : 27.9),
 (17.2 : 30.3), (15.7 : 28.1), (12 : 22.4), (6.5 : 15.1), (-2.2 : 11),
 (-6.6 : 4), (2,4,5,7,8,9,10,11,12)),
 ((-8.4 : 9), (-3.5 : 11.2), (-1.2 : 13.7), (1.7 : 16.4), (5.9 : 21.5), (9.7 : 24.8),
 (10.8 : 23.2), (10.2 : 20.8), (7.7 : 19.9), (1.4 : 18.7), (-4.9 : 11.4),
 (-7.5 : 8.8), (4,7,9)),

((8.6 : 14.4), (7.3 : 10.6), (10.7 : 15.6), (16.2 : 23.3), (22.4 : 28.9),
 (24.9 : 33.8), (26.2 : 33.9), (25.1 : 31.8), (23.9 : 31.1), (19.4 : 26.8),
 (14.1 : 23.8), (11.3 : 20.5), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-18.4 : -7.5), (-19.2 : -7.3), (-10.7 : 2.4), (-0.1 : 13.2), (6.8 : 18.5),
 (13.7 : 26.1), (17 : 26.5), (18.9 : 28.4), (10.3 : 21.4), (0.6 : 13.1),
 (-8.9 : 1.5), (-18.9 : -7.8), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((3.1 : 7.5), (3 : 7.6), (6.3 : 12), (14 : 22), (19.6 : 26.1), (23.4 : 30.2),
 (27.2 : 35.7), (25.5 : 32.6), (21.1 : 26.8), (16.9 : 23.6), (9.3 : 19),
 (5.4 : 14.5), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-0.3 : 7.6), (-0.3 : 6.9), (3.1 : 11.7), (10.5 : 21.1), (16.1 : 25.2),
 (21.1 : 28.8), (26.2 : 34.3), (23.2 : 30.8), (18.6 : 26.4), (13.7 : 22),
 (3.8 : 17.5), (0.9 : 10.8), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((11.5 : 17.7), (10.3 : 14.3), (12.1 : 17.3), (17.8 : 24.2), (23.8 : 30.7),
 (24.9 : 33.2), (25.8 : 33.5), (25.1 : 31.9), (24.2 : 31.5), (20.3 : 26.9),
 (14.8 : 24.3), (12.4 : 22.4), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-27.9 : -16), (-27.7 : -12.9), (-16.5 : -3.4), (-1.5 : 12), (4.5 : 17.5),
 (13.7 : 26.7), (16.1 : 25), (14.5 : 26), (6.9 : 19.2), (-2.6 : 10.9),
 (-14.4 : -3.6), (-26.1 : -13.8), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-2.5 : 3.2), (-2.7 : 4.2), (1.1 : 8), (7.4 : 15.3), (13.1 : 20.1), (18.1 : 24.1),
 (22.8 : 27.7), (22.9 : 28.4), (19.4 : 25.6), (13.8 : 20.3), (5.9 : 13.2),
 (-0.4 : 5.5), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-1.6 : 6), (-1.7 : 6), (1.9 : 10.5), (9.6 : 20.6), (15.2 : 24.6), (19.9 : 28.4),
 (25.1 : 32.8), (21.8 : 29.5), (17.2 : 26.1), (12 : 21.3), (3.8 : 16),
 (-0.3 : 8.5), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-22.7 : -11.5), (-21.1 : -9.6), (-11.5 : 0.7), (0.6 : 13.8), (7.8 : 18.9),
 (16.7 : 27.4), (19.1 : 26.5), (19.1 : 26.9), (10.5 : 19.8), (1.9 : 12.2),
 (-9 : 0.2), (-17.8 : -9.1), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((4 : 9.2), (2.8 : 7.9), (6 : 12.3), (12.4 : 21.7), (18 : 25.5), (22.5 : 29.9),
 (26.4 : 36.3), (24.5 : 32.8), (20.1 : 26.7), (15.8 : 23.9), (6.5 : 18.7),
 (2.8 : 14.6), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((2.4 : 9), (2 : 7.7), (4.4 : 11.2), (11.1 : 20.1), (16 : 23.7), (21.2 : 28),
 (25.9 : 33.8), (23.7 : 30.6), (20.1 : 26.5), (16 : 22.9), (6.5 : 17.1),
 (3 : 11.8), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((11.8 : 19.2), (12.2 : 17.7), (12.7 : 17.8), (16.4 : 22.7), (22.5 : 28.1),
 (24.9 : 31.5), (25.6 : 32.6), (24.9 : 30.8), (23.8 : 30), (20.4 : 27.3),
 (13.9 : 22.6), (10 : 19.7), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-11.9 : -1.9), (-12.8 : -2.7), (-5.2 : 6), (4 : 15.4), (11.7 : 23), (18.1 : 28.4),
 (22.2 : 29.7), (21.2 : 28.6), (14.4 : 24.1), (6 : 17.8), (-3.1 : 6.3),
 (-9.9 : -1.2), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-11.2 : 2.4), (-9 : 3.9), (-3.9 : 8.8), (3.7 : 19.3), (11.5 : 25), (14.9 : 29.1),
 (18.5 : 27.8), (17.2 : 25.9), (11.5 : 24.4), (5.2 : 18.5), (-4.3 : 11.6),
 (-8.7 : 3.6), (2,4,5,6,7,8,9,10,11,12)),
 ((2.3 : 16.9), (4 : 18.9), (6.4 : 21.3), (9.9 : 24.3), (15 : 22.6), (17.2 : 24.1),
 (17.4 : 22.8), (17.4 : 22.5), (16.3 : 23.2), (14.5 : 23.5), (9.4 : 20),
 (4.7 : 17.8), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-6.5 : 2), (-5.2 : 3.9), (0.5 : 9.5), (9.3 : 20.7), (15.2 : 25.6), (20.5 : 30.6),
 (22.6 : 30.1), (22 : 29.5), (17.5 : 26.9), (10.7 : 20.7), (2.4 : 12.6),
 (-3.4 : 4.9), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-5.2 : 5.3), (-3.8 : 4.4), (0 : 7.7), (6.6 : 19.5), (12.3 : 23.7), (15.9 : 27.8),
 (18.6 : 28.2), (17.1 : 26.2), (13.2 : 21), (8.1 : 15.5), (-0.6 : 11.3),
 (-3.5 : 4.5), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((6.9 : 12.9), (5.7 : 10.8), (7.6 : 12.8), (12.7 : 20.8), (18.1 : 24), (22.2 : 28.7),
 (26.2 : 34.6), (25.3 : 32.5), (21.6 : 28.1), (17.6 : 25.1), (9.3 : 19.8),
 (6.2 : 16.6), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((0.6 : 7.3), (1.3 : 8.2), (4.8 : 12.3), (12.5 : 22.7), (18.5 : 26.5), (22.4 : 30.3),
 (26.6 : 34.8), (24.3 : 31.6), (19.3 : 27.1), (13.7 : 22.4), (6.3 : 19.3),
 (3.3 : 11.9), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-15.6 : -7.6), (-19.4 : -9.8), (-7.3 : 0.2), (4.5 : 16.8), (8.4 : 18.9),
 (14.9 : 27.3), (17.4 : 30), (15.8 : 28.2), (12.6 : 24.7), (2.5 : 12.5),
 (-3.9 : 2.7), (-9.8 : -1.3), (1,3,5,8,10,11,12)),
 ((10 : 17.7), (9.1 : 14.7), (10.4 : 16.2), (15.8 : 23.9), (22.2 : 30.3),
 (23.9 : 34.3), (24.2 : 33.8), (23.4 : 32.1), (22.3 : 31.8), (19.2 : 27.6),
 (12.4 : 23.2), (9.8 : 21.6), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((10.4 : 18.5), (9.7 : 16.3), (10.2 : 16.1), (14.5 : 21.4), (20.3 : 26.2),
 (24.1 : 31.2), (25.2 : 33), (24.6 : 31.8), (22.8 : 29.1), (20.1 : 26.4),
 (14.2 : 21.4), (10.3 : 19), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-3.5 : 5.9), (-1.8 : 6.1), (1.3 : 9.6), (8.6 : 21.3), (15 : 25.7), (19.4 : 32.5),

(21.4 : 29.9), (20.6 : 29.3), (16.6 : 24.9), (10.9 : 18.1), (2.8 : 15.1),
 (-2.5 : 6.3), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((5.3 : 18.5), (7.9 : 22.1), (9.9 : 24.9), (11.7 : 25.1), (17.6 : 29.7),
 (18.4 : 28.9), (19.8 : 29), (18.7 : 26.9), (15.4 : 22.4), (14.3 : 21.7),
 (9.8 : 17.2), (7.1 : 18), (2,4,6,7,9,10,11,12)),
 ((-12.1 : 2.7), (-9.8 : 3.2), (-4.3 : 6.5), (1.8 : 15.7), (7.6 : 19.4),
 (10.3 : 23.7), (12.6 : 25.4), (11.8 : 24.3), (8.2 : 18.5), (3.2 : 11.6),
 (-3.9 : 9.2), (-9.3 : 6), (2,4,5,7,8,9,12)),
 ((-2.7 : 5.9), (-2 : 7.1), (1.4 : 11.5), (9.5 : 21.7), (15.7 : 25.7), (20.5 : 31.3),
 (25 : 33.5), (21.7 : 30.1), (16.6 : 27), (11.4 : 21.2), (2.5 : 16.3),
 (-1.1 : 8.5), (1,2,3,4,5,6,7,8,9,11,12)),
 ((-2.5 : 3.1), (-3.3 : 2.4), (0.8 : 7.6), (8.5 : 17.5), (14.9 : 23.5), (18.6 : 27.1),
 (21.5 : 27.5), (22.2 : 27.3), (19.1 : 24.9), (13.8 : 19.9), (6.5 : 11.9),
 (0.1 : 4.8), (1,2,3,4,7,9,11,12)),
 ((-13.1 : -1.4), (-16.3 : -4.5), (-3.7 : 7), (6.8 : 21.9), (9.5 : 22), (14.7 : 29),
 (16.6 : 30.8), (14.3 : 28.8), (10.8 : 25.9), (2.6 : 17.7), (0 : 12.4),
 (-4.7 : 5.7), (1,2,3,4,5,7,8,9,10,11,12)),
 ((2.3 : 8.1), (2.5 : 8.7), (5.2 : 13.2), (12.8 : 24.5), (18 : 26.7), (21.7 : 30.2),
 (25.7 : 35.4), (23.6 : 31.2), (19.6 : 27), (14.1 : 21.9), (9.3 : 19.6),
 (4.2 : 11.6), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-12.8 : -0.6), (-10.3 : 1.9), (-4.2 : 7.2), (3.4 : 18.3), (11.2 : 23.3),
 (15.3 : 27.8), (17.9 : 29.4), (16 : 27), (10.4 : 23.5), (4.2 : 15.9),
 (-3.3 : 8.6), (-9.3 : 1.1), (1,2,4,5,7,8,9,10,11,12)),
 ((8.7 : 17), (8.2 : 14.8), (9.7 : 16.5), (13.2 : 22.8), (20.6 : 28.3), (22.3 : 33),
 (24.4 : 36.9), (23.5 : 34.6), (21.4 : 29.1), (17.2 : 26.4), (9.2 : 20.6),
 (4.7 : 18.6), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-14.9 : -0.4), (-12.1 : 1.5), (-6 : 6.1), (2.1 : 17.2), (9.6 : 24.1),
 (13.4 : 27.8), (17 : 27.3), (15.5 : 25.6), (9.3 : 23.9), (2.9 : 16.4),
 (-7.6 : 9.1), (-12.7 : 0.2), (2,4,5,7,8,9,11,12)),
 ((-15.4 : 0.1), (-12.4 : -0.3), (-6.2 : 7.2), (2.3 : 18.3), (8.2 : 22.2),
 (11.5 : 26.4), (14 : 30.2), (13.5 : 28.1), (8.6 : 23.7), (1.8 : 15.1),
 (-5.8 : 9.3), (-12 : 3.3), (1,2,3,4,5,7,8,9,10,11,12)),
 ((15.8 : 21.5), (14.6 : 19.4), (14.4 : 18.8), (19.2 : 24.2), (26 : 31.7),
 (26.2 : 32.3), (27.1 : 33), (25.6 : 31.1), (25.7 : 31.9), (22 : 27.2),
 (16.1 : 23.2), (14.5 : 21.8), (1,2,3,4,5,6,7,8,9,10,11,12)),
 ((-3.5 : 7), (-3.2 : 7.1), (1.6 : 11.8), (9.1 : 22.3), (15.2 : 26.4), (20.5 : 33.3),
 (22.8 : 31.2), (21.5 : 29.3), (16 : 27), (10.7 : 21), (2.1 : 17.1),
 (-2.5 : 8.3), (2,3,4,5,6,7,8,9,10,11,12)),
 ((2.7 : 8.4), (2.7 : 8.7), (6.2 : 13), (12.1 : 23.5), (17.8 : 25.7), (20.2 : 30.3),
 (23.6 : 33), (22.5 : 31.2), (18.8 : 26.7), (14.4 : 22.7), (8.2 : 20),
 (5.1 : 13.3), (1,2,3,4,5,6,7,8,9,10,11,12))
))
 END

Fichier "chavent.txt"

60

5

"AnQing_1988"	2	2	2	2
"BaoDing_1988"	2	2	2	2
"BeiJing_1988"	2	2	2	2
"BoKeTu_1988"	1	1	1	1
"ChangChun_1988"	1	1	1	1
"ChangSha_1988"	2	2	2	2
"ChengDu_1988"	2	2	2	2
"ChongQing_1988"	2	3	3	5
"DaLian_1988"	2	2	2	2
"FuZhou_1988"	2	3	3	5
"GuangZhou_1988"	2	3	3	5
"GuiYang_1988"	2	2	2	2
"HaErBin_1988"	1	1	1	1
"HaiKou_1988"	2	3	3	5
"Hailaer_1988"	1	1	1	1
"HaMi_1988"	1	1	4	4
"HangZhou_1988"	2	2	2	2
"HanZhong_1988"	2	2	2	2
"HuHeHaoTe_1988"	1	1	4	4

"JiNan_1988"	2	2	2	2
"Jiu Quan_1988"	1	1	4	4
"KunMing_1988"	2	3	3	3
"LanZhou_1988"	1	1	4	4
"LaSa_1988"	1	1	4	4
"LiuZhou_1988"	2	3	3	5
"MuDanJiang_1988"	1	1	1	1
"NanChang_1988"	2	2	2	2
"NanJing_1988"	2	2	2	2
"NanNing_1988"	2	3	3	5
"NenJiang_1988"	1	1	1	1
"QingDao_1988"	2	2	2	2
"QingJiang_1988"	2	2	2	2
"QiQiHaEr_1988"	1	1	1	1
"QuZhou_1988"	2	2	2	2
"ShangHai_1988"	2	2	2	2
"ShanTou_1988"	2	3	3	5
"ShenYang_1988"	1	1	4	4
"TaiYuan_1988"	1	1	4	4
"TengChong_1988"	2	3	3	3
"TianJin_1988"	2	2	2	2
"TianShui_1988"	1	1	4	4
"WenZhou_1988"	2	3	3	5
"WuHan_1988"	2	2	2	2
"WuLuMuQi_1988"	1	1	4	4
"WuZhou_1988"	2	3	3	5
"XiaMen_1988"	2	3	3	5
"XiAn_1988"	2	2	2	2
"XiChang_1988"	2	3	3	3
"XiNing_1988"	1	1	4	4
"XuZhou_1988"	2	2	2	2
"YanTai_1988"	2	2	2	2
"Yi Ning_1988"	1	1	4	4
"YiChang_1988"	2	2	2	2
"YinChuan_1988"	1	1	4	4
"YongAn_1988"	2	3	3	5
"YuLin_1988"	1	1	4	4
"ZhangYe_1988"	1	1	4	4
"ZhanJiang_1988"	2	3	3	5
"ZhengZhou_1988"	2	2	2	2
"ZhiJiang_1988"	2	2	2	2

Fichier "scluster.txt"

60

5

"AnQing_1988"	1	2	2	2
"BaoDing_1988"	1	2	2	4
"BeiJing_1988"	1	2	2	4
"BoKeTu_1988"	2	3	3	1
"ChangChun_1988"	2	3	3	1
"ChangSha_1988"	1	2	2	2
"ChengDu_1988"	1	2	2	2
"ChongQing_1988"	1	2	2	2
"DaLian_1988"	1	2	2	4
"FuZhou_1988"	1	1	4	5
"GuangZhou_1988"	1	1	4	5
"GuiYang_1988"	1	2	2	2
"HaErBin_1988"	2	3	3	1
"HaiKou_1988"	1	1	4	5
"Hailaer_1988"	2	3	3	1
"HaMi_1988"	2	3	1	3
"HangZhou_1988"	1	2	2	2
"HanZhong_1988"	1	2	2	4
"HuHeHaoTe_1988"	2	3	1	3
"JiNan_1988"	1	2	2	4
"Jiu Quan_1988"	2	3	1	3

"KunMing_1988"	1	2	2	2
"LanZhou_1988"	2	3	1	3
"LaSa_1988"	2	3	1	3
"LiuZhou_1988"	1	1	4	5
"MuDanJiang_1988"	2	3	3	1
"NanChang_1988"	1	2	2	2
"NanJing_1988"	1	2	2	2
"NanNing_1988"	1	1	4	5
"NenJiang_1988"	2	3	3	1
"QingDao_1988"	1	2	2	4
"QingJiang_1988"	1	2	2	4
"QiQiHaEr_1988"	2	3	3	1
"QuZhou_1988"	1	2	2	2
"ShangHai_1988"	1	2	2	2
"ShanTou_1988"	1	1	4	5
"ShenYang_1988"	2	3	1	3
"TaiYuan_1988"	2	3	1	3
"TengChong_1988"	1	2	2	2
"TianJin_1988"	1	2	2	4
"TianShui_1988"	2	2	1	4
"WenZhou_1988"	1	2	2	2
"WuHan_1988"	1	2	2	2
"WuLuMuQi_1988"	2	3	3	3
"WuZhou_1988"	1	1	4	5
"XiaMen_1988"	1	1	4	5
"XiAn_1988"	1	2	2	4
"XiChang_1988"	1	1	4	5
"XiNing_1988"	2	3	1	3
"XuZhou_1988"	1	2	2	4
"YanTai_1988"	1	2	2	4
"Yi Ning_1988"	2	3	1	3
"YiChang_1988"	1	2	2	2
"YinChuan_1988"	2	3	1	3
"YongAn_1988"	1	1	4	5
"YuLin_1988"	2	3	1	3
"ZhangYe_1988"	2	3	1	3
"ZhanJiang_1988"	1	1	4	5
"ZhengZhou_1988"	1	2	2	4
"ZhiJiang_1988"	1	2	2	2

Annexe D : Données artificielles

Fichier "liste.txt"

Classes	Individus	Var1	Var2	Var3	Var4	Var5		
Cla_1	Ind_ 1	0.383374		1.110965		0.590228	0.262642	0.351802
Cla_1	Ind_ 1	0.618946		1.185339		0.950255	1.247932	1.286035
Cla_1	Ind_ 2	0.249840		0.530564		0.883816	0.371899	0.367458
Cla_1	Ind_ 2	1.107410		0.539140		1.369701	1.174078	0.815470
Cla_1	Ind_ 3	0.653523		0.765572		1.043092	1.347224	0.304254
Cla_1	Ind_ 3	1.626789		1.556185		1.083468	1.415586	1.289331
Cla_1	Ind_ 4	0.565645		0.242012		0.691916	1.028047	0.866909
Cla_1	Ind_ 4	0.583651		0.809259		1.340037	1.413099	0.893368
Cla_1	Ind_ 5	1.296350		1.088382		1.150197	0.397961	0.470855
Cla_1	Ind_ 5	1.506256		1.634266		1.778390	1.347423	0.991684
Cla_1	Ind_ 6	0.284143		0.668661		1.066530	0.766762	0.277184
Cla_1	Ind_ 6	1.181509		1.080813		1.422040	0.985763	0.865093
Cla_1	Ind_ 7	0.959181		1.304987		1.032548	0.817667	0.292840
Cla_1	Ind_ 7	1.390591		1.338618		1.289026	1.667791	1.048845
Cla_1	Ind_ 8	0.728599		0.388043		0.722449	0.166112	1.106372
Cla_1	Ind_ 8	1.131840		0.897794		1.208975	1.001556	1.655828
Cla_1	Ind_ 9	0.960738		0.502045		0.443419	0.144017	0.434553
Cla_1	Ind_ 9	1.056932		1.219352		0.571169	1.027131	0.749962
Cla_1	Ind_ 10	0.800195		0.812296		1.079928	0.411237	1.145589
Cla_1	Ind_ 10	1.453078		1.165517		1.505997	0.626850	1.537141

Cla_1	Ind_	11	1.050829	1.093005	0.584094	0.516434	0.607883
Cla_1	Ind_	11	1.424253	1.814341	1.369243	1.258736	1.302911
Cla_1	Ind_	12	0.523804	0.224403	0.217566	0.995821	0.907422
Cla_1	Ind_	12	0.891919	0.993194	0.789697	1.594394	1.089557
Cla_1	Ind_	13	0.980224	0.569597	0.494568	0.819315	1.487686
Cla_1	Ind_	13	1.089389	1.348766	1.387906	1.799997	1.500961
Cla_1	Ind_	14	0.411100	1.171194	0.033448	0.294794	0.412900
Cla_1	Ind_	14	1.301019	1.465453	1.026124	0.938551	0.695013
Cla_1	Ind_	15	0.707938	0.993149	1.097995	1.205252	0.545991
Cla_1	Ind_	15	1.135838	1.887219	1.533280	1.237663	1.006211
Cla_1	Ind_	16	0.594089	0.790582	0.817011	0.840098	0.280099
Cla_1	Ind_	16	1.061876	0.826685	1.205908	1.190878	1.169195
Cla_1	Ind_	17	0.324412	0.927839	0.226310	0.451476	0.383541
Cla_1	Ind_	17	1.198798	1.796701	0.957981	1.086932	0.733039
Cla_1	Ind_	18	0.932447	0.902036	0.685446	0.370617	0.317713
Cla_1	Ind_	18	1.913800	1.313272	0.707389	1.019288	0.892682
Cla_1	Ind_	19	0.518540	0.769616	1.205603	0.852123	1.386181
Cla_1	Ind_	19	1.021882	1.130985	1.393811	1.662572	1.567064
Cla_1	Ind_	20	1.047456	0.437239	1.087512	0.991791	1.076693
Cla_1	Ind_	20	1.796716	1.117161	1.845775	1.046693	1.550859
Cla_1	Ind_	21	0.166860	0.673803	1.328913	1.099857	0.784631
Cla_1	Ind_	21	0.989395	1.295160	1.507447	1.152745	1.402844
Cla_1	Ind_	22	0.985839	1.040315	0.327616	0.518387	0.562716
Cla_1	Ind_	22	1.568651	1.716025	1.218055	0.832148	0.841105
Cla_1	Ind_	23	0.373074	0.534471	0.295053	0.359981	0.446150
Cla_1	Ind_	23	1.368343	1.099765	1.262337	1.254357	1.235939
Cla_1	Ind_	24	0.466750	0.914426	0.313868	0.830485	0.563585
Cla_1	Ind_	24	1.206305	1.553697	1.203848	1.249931	1.257729
Cla_1	Ind_	25	0.535447	0.718436	0.941511	0.990005	0.475143
Cla_1	Ind_	25	0.744774	1.063692	1.022507	1.478851	1.292520
Cla_1	Ind_	26	0.914777	0.744316	0.482528	0.319803	0.793207
Cla_1	Ind_	26	1.848460	1.308009	1.230140	1.249916	1.312784
Cla_1	Ind_	27	0.271233	0.507614	0.848994	1.107700	0.951796
Cla_1	Ind_	27	0.889294	0.576128	0.881375	1.488845	1.931745
Cla_1	Ind_	28	0.732093	0.771645	1.102573	0.755455	0.306574
Cla_1	Ind_	28	0.791299	1.715461	1.160466	1.734764	1.303308
Cla_1	Ind_	29	0.275170	0.952330	0.646519	0.961104	0.386608
Cla_1	Ind_	29	1.034959	1.775689	1.450804	1.073962	0.804804
Cla_1	Ind_	30	0.513321	0.851924	0.913221	0.146413	0.888318
Cla_1	Ind_	30	1.295907	1.229835	1.417670	1.037248	1.483489
Cla_1	Ind_	31	0.941069	0.588870	0.094867	0.303430	0.718604
Cla_1	Ind_	31	1.803247	1.256737	1.094470	1.125965	1.505432
Cla_1	Ind_	32	1.155522	0.071017	0.297586	0.929411	0.867733
Cla_1	Ind_	32	1.815851	0.929044	1.198401	1.732749	1.751701
Cla_1	Ind_	33	0.323756	0.724509	1.276208	0.725700	0.935804
Cla_1	Ind_	33	1.261467	1.462844	1.339930	1.222144	1.695532
Cla_1	Ind_	34	0.567019	0.749535	0.945585	1.060045	1.064272
Cla_1	Ind_	34	1.505524	0.974181	1.391736	1.270928	1.711478
Cla_1	Ind_	35	0.770562	0.742882	0.468245	0.650884	0.145817
Cla_1	Ind_	35	0.846644	0.875179	0.953642	0.676519	0.941221
Cla_1	Ind_	36	0.928205	0.412595	1.209128	0.335185	0.315271
Cla_1	Ind_	36	1.618229	1.063555	1.557100	1.161351	1.215812
Cla_1	Ind_	37	0.299249	1.133244	1.031663	0.441939	1.174947
Cla_1	Ind_	37	1.139729	1.857112	1.413694	1.168310	1.243187
Cla_1	Ind_	38	0.706503	0.553957	0.603046	1.025559	0.790643
Cla_1	Ind_	38	1.152715	1.129841	1.074526	1.697027	1.754448
Cla_1	Ind_	39	0.551775	0.696249	0.387616	0.327616	0.523240
Cla_1	Ind_	39	0.725089	1.262703	0.764183	0.806604	1.495285
Cla_1	Ind_	40	0.380383	0.772103	1.154454	0.644948	0.545930
Cla_1	Ind_	40	1.358959	1.762917	1.324717	1.450240	0.792886
Cla_1	Ind_	41	0.400723	1.129719	0.914686	1.098407	1.254799
Cla_1	Ind_	41	1.119068	1.169820	1.085833	1.241356	1.356761
Cla_1	Ind_	42	0.573412	0.606357	0.553880	0.936460	1.018693
Cla_1	Ind_	42	0.960723	1.092578	1.508835	1.525864	1.923688
Cla_1	Ind_	43	0.537645	0.947417	0.842692	0.776238	1.150136
Cla_1	Ind_	43	1.502182	1.346202	1.567263	1.179235	1.271722
Cla_1	Ind_	44	1.053545	0.151524	0.811533	0.901685	0.919523

Cla_1	Ind_	44	1.206320	1.115696	0.967238	1.804910	0.937559
Cla_1	Ind_	45	1.120899	0.933927	0.471786	0.211188	1.199866
Cla_1	Ind_	45	1.239921	1.808557	0.911100	0.914548	1.620380
Cla_1	Ind_	46	0.798944	0.778985	0.871029	0.462589	1.222434
Cla_1	Ind_	46	1.104892	1.202002	1.106662	0.664541	1.675939
Cla_1	Ind_	47	0.810907	0.588855	1.294748	0.807718	0.766976
Cla_1	Ind_	47	0.925321	1.583728	1.559679	1.233818	1.736549
Cla_1	Ind_	48	0.585055	1.041993	0.885006	0.513413	1.206412
Cla_1	Ind_	48	0.959395	1.762871	1.795282	1.407697	1.272271
Cla_1	Ind_	49	0.271645	0.763283	1.350841	0.833979	0.996445
Cla_1	Ind_	49	0.928983	1.031358	1.473983	1.803827	1.275811
Cla_1	Ind_	50	1.294870	1.094256	0.604205	0.160655	1.252205
Cla_1	Ind_	50	1.554430	1.817728	1.334605	0.866822	1.642750
Cla_1	Ind_	51	0.965835	0.441847	0.889279	0.517533	0.299722
Cla_1	Ind_	51	1.689795	1.337535	1.451338	1.402722	1.168615
Cla_1	Ind_	52	0.905759	0.698904	1.084689	0.559984	0.347072
Cla_1	Ind_	52	1.608631	1.442305	1.368419	1.556810	0.835429
Cla_1	Ind_	53	0.322001	0.373501	0.670202	0.380001	0.993011
Cla_1	Ind_	53	0.731529	1.038621	0.714820	1.102985	1.142521
Cla_1	Ind_	54	0.857021	1.156514	0.664815	0.987106	0.500626
Cla_1	Ind_	54	1.351634	1.832928	1.336985	1.940504	0.597095
Cla_1	Ind_	55	1.102405	1.003830	0.481689	0.514222	0.916318
Cla_1	Ind_	55	1.639592	1.684820	1.260277	1.265893	1.048433
Cla_1	Ind_	56	0.749397	0.730232	0.916593	0.411939	0.524689
Cla_1	Ind_	56	0.833750	1.471526	1.337046	0.742119	1.267083
Cla_1	Ind_	57	0.548357	0.912366	0.700522	1.256996	0.612918
Cla_1	Ind_	57	1.481399	1.002457	1.193670	1.457108	1.474822
Cla_1	Ind_	58	1.081484	0.396039	0.523118	0.573290	0.381115
Cla_1	Ind_	58	1.828974	0.625568	0.996246	1.092257	1.357921
Cla_1	Ind_	59	0.132862	1.207999	0.975112	0.460265	0.895749
Cla_1	Ind_	59	0.906323	1.665410	1.044939	0.642338	1.884274
Cla_1	Ind_	60	0.975036	0.996536	0.786462	0.662420	0.549547
Cla_1	Ind_	60	1.939634	1.167440	1.221564	1.659154	0.931669
Cla_1	Ind_	61	0.997375	0.585376	0.858623	0.814386	0.380062
Cla_1	Ind_	61	1.908261	1.533250	1.291131	1.772881	0.757027
Cla_1	Ind_	62	1.092792	1.125797	0.218085	0.763756	0.619007
Cla_1	Ind_	62	1.263024	1.447249	0.931211	1.610370	1.131687
Cla_1	Ind_	63	0.401044	0.495682	0.470489	0.925764	0.809900
Cla_1	Ind_	63	0.985382	1.467605	1.074145	1.619694	0.975768
Cla_1	Ind_	64	1.201804	0.824107	0.415754	0.871029	1.056429
Cla_1	Ind_	64	1.728858	1.203940	1.129246	1.659963	1.478835
Cla_1	Ind_	65	1.261513	0.634632	1.014298	0.603076	1.256539
Cla_1	Ind_	65	1.682638	1.084140	1.347316	0.966582	1.435774
Cla_1	Ind_	66	0.585238	0.910413	0.179571	1.056307	0.354778
Cla_1	Ind_	66	1.466552	1.749001	0.932432	1.684286	1.291849
Cla_1	Ind_	67	1.113758	1.201178	0.729987	0.554247	0.517426
Cla_1	Ind_	67	1.249382	1.551134	0.934584	1.520676	0.615268
Cla_1	Ind_	68	0.256172	0.530229	0.570269	1.120792	1.014466
Cla_1	Ind_	68	0.812738	0.906552	1.408521	1.642079	1.341227
Cla_1	Ind_	69	0.566774	0.629627	1.109424	0.447752	0.793115
Cla_1	Ind_	69	1.398465	1.292825	1.737922	1.149983	1.089969
Cla_1	Ind_	70	0.676214	1.247993	0.363170	0.305002	1.060518
Cla_1	Ind_	70	0.817576	1.445235	1.335398	1.282235	1.595599
Cla_1	Ind_	71	0.585055	1.248833	0.951979	0.541398	0.970779
Cla_1	Ind_	71	0.899640	1.707709	1.575655	0.807794	1.726417
Cla_1	Ind_	72	0.931837	1.261086	0.600925	0.888959	0.549364
Cla_1	Ind_	72	1.859233	1.485122	0.740699	1.874447	1.170385
Cla_1	Ind_	73	1.173284	1.070818	0.589633	0.331568	0.092196
Cla_1	Ind_	73	1.492782	1.260704	1.387875	1.252922	1.038179
Cla_1	Ind_	74	1.072329	0.564211	0.697470	0.745643	0.767953
Cla_1	Ind_	74	1.226417	1.114948	1.330943	0.946150	1.338099
Cla_1	Ind_	75	0.792917	0.162908	0.359020	0.926023	0.834910
Cla_1	Ind_	75	0.854564	1.058962	0.791803	1.424909	0.940138
Cla_1	Ind_	76	1.061663	1.142201	0.797494	0.718940	0.578127
Cla_1	Ind_	76	1.778176	1.203604	1.669469	1.261254	0.709357
Cla_1	Ind_	77	0.587878	0.805658	1.148457	0.093463	0.583178
Cla_1	Ind_	77	0.603229	1.269936	1.222556	1.084216	1.383648

Cla_1	Ind_	78	1.047838	0.518952	0.343623	1.333491	0.109867
Cla_1	Ind_	78	1.539735	1.140248	0.713324	1.524049	0.944334
Cla_1	Ind_	79	0.748314	0.389996	0.598453	0.693335	0.419187
Cla_1	Ind_	79	1.076052	0.992218	1.142781	1.576418	0.608707
Cla_1	Ind_	80	0.902753	0.741295	0.554613	1.234336	0.756325
Cla_1	Ind_	80	1.844737	1.487564	0.567400	1.543123	1.603702
Cla_1	Ind_	81	0.589785	0.257118	0.283776	0.937925	0.110218
Cla_1	Ind_	81	1.363124	0.937468	1.056444	1.829524	0.996536
Cla_1	Ind_	82	0.735679	0.625309	1.268014	0.393689	0.673421
Cla_1	Ind_	82	0.882443	0.782479	1.565966	0.839900	1.498550
Cla_1	Ind_	83	0.676885	1.433485	0.954955	1.191336	0.921583
Cla_1	Ind_	83	1.333033	1.533372	1.310587	1.444334	1.756691
Cla_1	Ind_	84	0.833567	0.843669	0.850108	0.925428	0.737907
Cla_1	Ind_	84	1.759804	0.968917	1.540559	0.935560	1.149083
Cla_1	Ind_	85	1.068392	0.964675	0.667272	0.757195	1.332789
Cla_1	Ind_	85	1.633564	1.306970	0.919233	0.986908	1.434050
Cla_1	Ind_	86	1.231132	1.181539	0.626530	1.084399	0.666158
Cla_1	Ind_	86	1.437193	1.196188	1.384121	1.801218	1.210700
Cla_1	Ind_	87	1.013092	0.800974	0.376171	0.859081	0.985000
Cla_1	Ind_	87	1.402661	1.157430	0.894772	1.842448	1.969405
Cla_1	Ind_	88	0.363842	0.540071	0.536195	0.778588	0.747459
Cla_1	Ind_	88	0.648000	1.468627	0.769967	1.205451	0.977233
Cla_1	Ind_	89	0.673574	0.937773	0.825861	0.266152	0.438383
Cla_1	Ind_	89	1.173620	1.378063	1.319681	0.740013	0.797708
Cla_1	Ind_	90	1.007553	1.014817	0.462279	0.671834	0.894559
Cla_1	Ind_	90	1.160756	1.879894	1.135762	0.943785	1.821070
Cla_1	Ind_	91	1.206824	0.764092	1.310617	0.267769	1.066774
Cla_1	Ind_	91	1.709067	1.036592	1.644765	0.807978	1.719535
Cla_1	Ind_	92	0.456603	0.627232	0.442946	0.705710	0.964873
Cla_1	Ind_	92	0.993301	0.916425	1.291269	1.184393	1.178716
Cla_1	Ind_	93	0.778771	1.322016	0.635838	0.449507	0.668218
Cla_1	Ind_	93	1.247749	1.573244	0.946516	1.268471	1.028397
Cla_1	Ind_	94	0.145100	1.038179	1.215110	0.118305	0.555086
Cla_1	Ind_	94	0.938704	1.565081	1.759560	0.914167	1.441954
Cla_1	Ind_	95	1.342509	0.930692	0.857860	0.474303	1.120014
Cla_1	Ind_	95	1.399366	1.725913	1.629612	1.131245	1.630467
Cla_1	Ind_	96	1.163167	0.320490	0.294275	0.788461	0.722755
Cla_1	Ind_	96	1.511994	1.208274	1.211173	1.720130	0.863506
Cla_1	Ind_	97	0.756157	0.877483	0.629673	0.821558	0.154805
Cla_1	Ind_	97	1.170507	1.642033	0.807352	1.495987	1.138661
Cla_1	Ind_	98	0.871838	0.119541	1.043870	0.982818	0.837642
Cla_1	Ind_	98	1.677038	1.042207	1.100208	1.889737	1.836909
Cla_1	Ind_	99	1.070864	1.129673	0.731651	0.687201	1.088015
Cla_1	Ind_	99	1.357524	1.786157	0.845698	0.822031	1.177648
Cla_1	Ind_	100	0.570193	0.779504	0.436506	0.334880	1.016129
Cla_1	Ind_	100	0.963271	1.501358	1.362041	1.065157	1.659276
Cla_2	Ind_	101	0.219459	0.288934	0.845073	0.402310	-1.050462
Cla_2	Ind_	101	0.783044	1.097674	1.324946	1.298273	-0.303858
Cla_2	Ind_	102	0.244636	0.953734	0.796503	0.409177	-1.435636
Cla_2	Ind_	102	1.103580	1.467269	0.811487	0.773629	-1.269738
Cla_2	Ind_	103	1.265679	0.616749	0.319971	0.746071	-1.201819
Cla_2	Ind_	103	1.711371	0.621418	0.697851	1.317255	-0.594653
Cla_2	Ind_	104	0.334712	0.774728	0.253197	0.882015	-1.131092
Cla_2	Ind_	104	0.997757	1.126850	0.860881	1.684622	-0.829142
Cla_2	Ind_	105	1.012635	0.993042	0.968184	0.844417	-0.742561
Cla_2	Ind_	105	1.739311	1.918760	1.110523	1.079745	-0.532960
Cla_2	Ind_	106	0.857830	0.996948	0.915281	0.617573	-0.671728
Cla_2	Ind_	106	1.701483	1.996643	1.307718	0.914853	-0.647984
Cla_2	Ind_	107	0.829554	1.149098	0.049394	0.639439	-1.257683
Cla_2	Ind_	107	0.922178	1.205313	0.968184	0.912336	-0.566500
Cla_2	Ind_	108	0.974364	0.882260	1.009507	0.483383	-0.809702
Cla_2	Ind_	108	1.700858	1.087619	1.477966	1.432539	-0.701422
Cla_2	Ind_	109	0.906430	0.930525	0.891736	0.539003	-1.476287
Cla_2	Ind_	109	1.291665	1.539491	1.253075	0.764107	-0.673406
Cla_2	Ind_	110	0.522111	1.078768	0.340327	0.960128	-1.052644
Cla_2	Ind_	110	1.512101	1.424329	0.997635	1.023667	-0.547838
Cla_2	Ind_	111	0.172704	0.189016	1.041368	0.891369	-0.874981

Cla_2	Ind_ 111	1.122288	1.094134	1.344417	0.961745	-0.191794
Cla_2	Ind_ 112	0.214606	1.030656	0.602405	1.079562	-1.596408
Cla_2	Ind_ 112	1.091861	1.612705	0.780297	1.554827	-1.092486
Cla_2	Ind_ 113	1.029221	0.495209	0.804315	0.984268	-1.560884
Cla_2	Ind_ 113	1.434812	1.063952	1.560167	1.459563	-1.193075
Cla_2	Ind_ 114	1.317133	0.685522	0.873682	0.988647	-1.123447
Cla_2	Ind_ 114	1.352229	1.348506	0.978561	1.910031	-0.777459
Cla_2	Ind_ 115	0.784234	1.188543	0.820154	1.334071	-1.044450
Cla_2	Ind_ 115	1.159215	1.505417	1.092044	1.631870	-0.477172
Cla_2	Ind_ 116	0.315332	1.140614	0.555818	0.030152	-0.952666
Cla_2	Ind_ 116	1.076647	1.538270	1.445982	1.024781	-0.902158
Cla_2	Ind_ 117	0.934294	1.029664	1.058687	1.280496	-1.761635
Cla_2	Ind_ 117	1.128361	1.656423	1.256539	1.403821	-1.018509
Cla_2	Ind_ 118	0.343532	0.617924	0.273721	0.980621	-1.273370
Cla_2	Ind_ 118	1.284600	0.954237	1.006806	1.688620	-0.526154
Cla_2	Ind_ 119	0.680486	0.471114	0.879925	0.652776	-0.953917
Cla_2	Ind_ 119	0.824961	0.532121	1.732551	0.768380	-0.939665
Cla_2	Ind_ 120	0.386517	0.909146	1.008500	0.835749	-0.948958
Cla_2	Ind_ 120	0.841029	1.595294	1.080386	1.037690	-0.658605
Cla_2	Ind_ 121	0.820475	0.811579	1.048685	0.522614	-1.266640
Cla_2	Ind_ 121	1.052904	1.344157	1.208777	1.485656	-0.341838
Cla_2	Ind_ 122	0.521973	0.180761	0.458434	0.284921	-1.624470
Cla_2	Ind_ 122	0.857921	1.175939	1.456450	0.910092	-1.186743
Cla_2	Ind_ 123	1.407300	1.249611	0.342769	0.809732	-1.174001
Cla_2	Ind_ 123	1.455733	1.539628	1.111835	1.011704	-0.569857
Cla_2	Ind_ 124	0.718436	0.780465	0.940397	0.764321	-1.121815
Cla_2	Ind_ 124	1.184790	1.415189	1.769189	1.485229	-0.746681
Cla_2	Ind_ 125	0.315317	0.602588	0.648396	0.755470	-1.091037
Cla_2	Ind_ 125	1.053224	1.507675	0.837336	1.453978	-0.739738
Cla_2	Ind_ 126	0.954268	0.934721	0.774819	1.229972	-1.357830
Cla_2	Ind_ 126	1.034654	1.546770	1.465941	1.379086	-0.490097
Cla_2	Ind_ 127	1.104205	1.206076	0.679510	0.530412	-1.281137
Cla_2	Ind_ 127	1.718909	1.249290	1.656041	1.099612	-1.107212
Cla_2	Ind_ 128	0.174032	0.979064	0.491424	0.541047	-1.269021
Cla_2	Ind_ 128	1.043077	1.723380	0.818339	0.617649	-0.449019
Cla_2	Ind_ 129	0.820963	0.759606	0.714957	0.491134	-0.918943
Cla_2	Ind_ 129	1.269219	1.058351	1.216163	0.814173	-0.605060
Cla_2	Ind_ 130	0.847148	1.010834	0.328059	0.594363	-0.977325
Cla_2	Ind_ 130	1.806223	1.735862	1.272057	0.660100	-0.452742
Cla_2	Ind_ 131	0.631581	0.134449	0.992798	0.352275	-1.418058
Cla_2	Ind_ 131	1.587695	1.010086	1.314920	0.857326	-1.127766
Cla_2	Ind_ 132	1.144398	1.105533	0.802973	0.600131	-1.181921
Cla_2	Ind_ 132	1.695563	1.220069	1.182104	1.393400	-0.799554
Cla_2	Ind_ 133	0.922086	0.908689	0.151952	0.289087	-0.882504
Cla_2	Ind_ 133	1.454237	1.303873	0.859828	0.912152	-0.391003
Cla_2	Ind_ 134	0.998886	0.487426	0.874264	1.140294	-1.510544
Cla_2	Ind_ 134	1.495788	1.272790	1.231361	1.771142	-1.136296
Cla_2	Ind_ 135	0.259987	1.145726	0.362911	0.236305	-1.791910
Cla_2	Ind_ 135	1.003265	1.757714	0.692755	0.875515	-0.944838
Cla_2	Ind_ 136	1.065874	0.794656	0.451445	0.899335	-1.315409
Cla_2	Ind_ 136	1.662755	1.648625	0.577914	1.516480	-1.249458
Cla_2	Ind_ 137	0.356868	0.681280	0.537065	0.898953	-1.279015
Cla_2	Ind_ 137	0.980972	1.000687	1.198065	1.705832	-0.667913
Cla_2	Ind_ 138	0.847880	0.329722	0.775567	0.208945	-1.647740
Cla_2	Ind_ 138	1.748482	0.959899	1.029267	1.064135	-1.219932
Cla_2	Ind_ 139	0.922208	0.935270	0.756035	1.240684	-1.609806
Cla_2	Ind_ 139	1.224494	1.160833	0.866665	1.375393	-0.821696
Cla_2	Ind_ 140	1.000412	0.936186	0.432798	0.296701	-1.395337
Cla_2	Ind_ 140	1.790048	1.551408	1.289453	1.160283	-1.145787
Cla_2	Ind_ 141	0.549989	0.513092	0.811274	0.465270	-0.724845
Cla_2	Ind_ 141	1.534822	0.594516	1.238075	0.724067	-0.492080
Cla_2	Ind_ 142	0.584002	1.391583	0.517731	0.379055	-1.317789
Cla_2	Ind_ 142	0.709098	1.471694	0.576449	1.293756	-0.885006
Cla_2	Ind_ 143	1.027573	0.613178	0.994690	0.756996	-1.451811
Cla_2	Ind_ 143	1.864757	1.455275	1.392376	0.761330	-0.497009
Cla_2	Ind_ 144	0.778146	0.957991	0.711371	0.490387	-1.963317
Cla_2	Ind_ 144	1.019242	1.213126	1.656652	1.380612	-1.022340

Cla_2	Ind_ 145	0.708472	1.005463	0.530824	0.295312	-1.747719
Cla_2	Ind_ 145	1.494629	1.147893	0.913831	0.713233	-1.087786
Cla_2	Ind_ 146	1.322672	0.980026	0.790139	0.687323	-1.701987
Cla_2	Ind_ 146	1.387524	1.642094	1.592837	1.372951	-1.012482
Cla_2	Ind_ 147	0.839030	0.096728	0.791238	0.309458	-0.581484
Cla_2	Ind_ 147	1.616764	0.965407	1.497803	0.861446	-0.522706
Cla_2	Ind_ 148	0.702383	1.171789	0.331004	0.520264	-1.359325
Cla_2	Ind_ 148	0.847560	1.791772	1.253487	1.214805	-1.203406
Cla_2	Ind_ 149	0.479827	1.349071	1.181784	0.189082	-1.815409
Cla_2	Ind_ 149	1.001267	1.455489	1.623508	0.971160	-0.841014
Cla_2	Ind_ 150	0.840678	0.482330	0.128437	0.411695	-1.685171
Cla_2	Ind_ 150	1.711066	0.938948	0.879070	0.816401	-0.692557
Cla_2	Ind_ 151	0.412534	0.565432	0.343348	0.915830	-1.694174
Cla_2	Ind_ 151	0.664617	0.813120	0.963668	1.854915	-0.914518
Cla_2	Ind_ 152	0.817179	1.181417	1.089206	0.332591	-1.141484
Cla_2	Ind_ 152	1.473937	1.636876	1.753380	0.968932	-0.719443
Cla_2	Ind_ 153	1.172735	0.810343	0.556810	0.831950	-0.803003
Cla_2	Ind_ 153	1.713309	1.346797	0.953764	0.868572	-0.606494
Cla_2	Ind_ 154	0.375576	1.055284	0.619724	0.917173	-1.244377
Cla_2	Ind_ 154	0.764992	1.126209	0.775613	1.371502	-0.547029
Cla_2	Ind_ 155	0.599063	0.478668	0.992416	1.011460	-1.602023
Cla_2	Ind_ 155	1.283532	1.314386	1.192007	1.887463	-0.614566
Cla_2	Ind_ 156	0.237449	0.853420	0.773080	0.406842	-1.443617
Cla_2	Ind_ 156	1.133168	1.295480	1.481597	0.692404	-1.035981
Cla_2	Ind_ 157	1.040223	0.780419	0.694861	1.249565	-1.397183
Cla_2	Ind_ 157	1.750420	1.676473	1.092273	1.558168	-0.826609
Cla_2	Ind_ 158	0.486709	0.877163	1.169774	0.563875	-1.275124
Cla_2	Ind_ 158	1.220405	1.613544	1.309091	0.836467	-0.363994
Cla_2	Ind_ 159	0.609012	0.155660	0.444090	0.935575	-0.857570
Cla_2	Ind_ 159	1.126667	1.063097	0.962050	1.374096	-0.767144
Cla_2	Ind_ 160	0.541673	0.350261	0.458586	0.445250	-1.770028
Cla_2	Ind_ 160	0.617450	0.705344	1.155873	0.840098	-1.094287
Cla_2	Ind_ 161	1.125584	0.886166	0.882900	0.325617	-1.937666
Cla_2	Ind_ 161	1.324290	1.501877	1.426893	0.870983	-0.963088
Cla_2	Ind_ 162	0.479049	0.962386	0.833522	0.677801	-1.854930
Cla_2	Ind_ 162	1.449019	1.493347	1.526322	1.066424	-1.045152
Cla_2	Ind_ 163	0.639149	0.418317	0.551515	0.980087	-1.314066
Cla_2	Ind_ 163	0.699973	1.202795	0.873424	0.989242	-0.806207
Cla_2	Ind_ 164	0.726676	0.726173	0.871487	0.624668	-1.727958
Cla_2	Ind_ 164	1.485214	1.075976	1.813807	1.621952	-1.165105
Cla_2	Ind_ 165	1.170171	0.664296	0.220420	0.881878	-1.696814
Cla_2	Ind_ 165	1.556932	0.948515	0.832987	0.900494	-1.241111
Cla_2	Ind_ 166	0.889554	0.538255	0.707404	1.084124	-0.944487
Cla_2	Ind_ 166	1.826319	1.265542	1.515778	1.666723	-0.865474
Cla_2	Ind_ 167	1.062075	0.795907	0.661168	1.000961	-1.623432
Cla_2	Ind_ 167	1.524277	1.385342	1.477233	1.344539	-1.356120
Cla_2	Ind_ 168	0.725517	1.085147	1.311441	0.911115	-0.697562
Cla_2	Ind_ 168	0.957610	1.456893	1.428297	1.540040	-0.514542
Cla_2	Ind_ 169	0.584567	0.797479	0.246590	0.820719	-1.096362
Cla_2	Ind_ 169	1.486572	1.762688	1.148137	1.774117	-0.206015
Cla_2	Ind_ 170	1.035981	0.984436	0.384548	1.202643	-1.026582
Cla_2	Ind_ 170	1.221442	1.249428	1.116749	1.373577	-0.545640
Cla_2	Ind_ 171	0.178213	1.293023	0.520814	0.591952	-1.672018
Cla_2	Ind_ 171	0.984756	1.417661	1.094134	1.353999	-0.944243
Cla_2	Ind_ 172	1.487350	0.801813	0.638081	0.436659	-0.939405
Cla_2	Ind_ 172	1.502701	1.493454	0.710196	1.113285	-0.382748
Cla_2	Ind_ 173	0.481475	0.413404	0.629368	1.330164	-1.861721
Cla_2	Ind_ 173	1.260979	1.051637	0.833964	1.611911	-1.092502
Cla_2	Ind_ 174	0.628346	0.334956	0.972259	0.706305	-1.681921
Cla_2	Ind_ 174	0.675314	0.683538	1.315500	1.122822	-1.136738
Cla_2	Ind_ 175	1.030274	0.719398	0.750465	0.321100	-1.453413
Cla_2	Ind_ 175	1.368267	1.595645	1.320490	0.786874	-1.026612
Cla_2	Ind_ 176	1.120045	0.304529	0.820170	0.938948	-1.639180
Cla_2	Ind_ 176	1.606632	0.870342	1.598636	1.789499	-0.774804
Cla_2	Ind_ 177	0.771493	0.945830	0.917753	1.151555	-1.113651
Cla_2	Ind_ 177	1.518342	1.711447	1.826075	1.673513	-1.006012
Cla_2	Ind_ 178	0.442488	0.842799	0.172124	1.241691	-1.603015

Cla_2	Ind_ 178	1.265542	1.012635	0.829585	1.305414	-1.217353
Cla_2	Ind_ 179	0.780084	0.582659	0.375393	0.029344	-0.879192
Cla_2	Ind_ 179	1.587695	0.725517	0.721534	1.007767	-0.825541
Cla_2	Ind_ 180	1.095416	0.921186	0.548617	0.763070	-1.890194
Cla_2	Ind_ 180	1.732867	0.942244	0.938566	1.544954	-1.064150
Cla_2	Ind_ 181	0.321299	0.558657	0.856029	0.409619	-1.397763
Cla_2	Ind_ 181	0.955565	1.542146	1.070818	1.191778	-0.917158
Cla_2	Ind_ 182	1.056764	1.106525	0.422681	0.712256	-1.045320
Cla_2	Ind_ 182	1.159276	1.862346	0.879086	1.711005	-0.403668
Cla_2	Ind_ 183	0.648152	1.043260	0.921964	0.692053	-1.526902
Cla_2	Ind_ 183	1.594256	1.333064	1.212348	1.404416	-0.980224
Cla_2	Ind_ 184	0.516602	0.787240	0.900494	0.925993	-0.613392
Cla_2	Ind_ 184	1.468657	1.552400	1.346248	1.671529	-0.408765
Cla_2	Ind_ 185	0.674566	0.632099	0.760842	0.505051	-0.651067
Cla_2	Ind_ 185	1.584384	1.550493	1.250237	1.394391	-0.468139
Cla_2	Ind_ 186	0.650456	0.759926	1.104404	1.124088	-1.215888
Cla_2	Ind_ 186	0.853465	1.701056	1.145756	1.158605	-0.671407
Cla_2	Ind_ 187	0.915647	1.432157	1.075732	0.342677	-1.325541
Cla_2	Ind_ 187	0.977020	1.559908	1.126087	1.002915	-1.009583
Cla_2	Ind_ 188	0.948317	0.816691	1.095294	0.743721	-0.914808
Cla_2	Ind_ 188	1.358852	1.038316	1.261742	1.251366	-0.479736
Cla_2	Ind_ 189	0.334773	0.811609	0.960877	1.069475	-1.106647
Cla_2	Ind_ 189	0.674261	0.940184	1.693976	1.712806	-0.592166
Cla_2	Ind_ 190	0.865383	1.224174	0.807169	0.994049	-1.099551
Cla_2	Ind_ 190	1.602557	1.690008	0.919355	1.815271	-0.990875
Cla_2	Ind_ 191	0.988754	0.330790	1.402249	0.559221	-1.676640
Cla_2	Ind_ 191	1.750130	0.747459	1.470000	0.770501	-1.087527
Cla_2	Ind_ 192	0.309214	0.734947	0.773736	1.020478	-0.880230
Cla_2	Ind_ 192	0.927488	1.425245	1.689383	1.318979	-0.502045
Cla_2	Ind_ 193	1.111240	0.619129	0.797647	0.983169	-0.634983
Cla_2	Ind_ 193	1.883755	0.789697	0.978866	1.178793	-0.444731
Cla_2	Ind_ 194	0.729728	0.521744	0.160649	0.820521	-1.722541
Cla_2	Ind_ 194	1.033723	0.719901	0.931822	1.701559	-1.273919
Cla_2	Ind_ 195	0.525605	0.713919	1.130131	1.013031	-1.024888
Cla_2	Ind_ 195	1.327143	1.208777	1.368633	1.105777	-0.131062
Cla_2	Ind_ 196	1.327204	0.674871	0.869182	0.300790	-0.807825
Cla_2	Ind_ 196	1.386837	1.306909	1.625248	1.210700	-0.608203
Cla_2	Ind_ 197	1.027223	0.626988	1.192129	0.955061	-1.023560
Cla_2	Ind_ 197	1.078463	1.417356	1.216178	1.508423	-0.148015
Cla_2	Ind_ 198	1.144215	0.653996	0.226800	0.116565	-0.853679
Cla_2	Ind_ 198	1.449644	1.135884	0.878567	0.885144	-0.633946
Cla_2	Ind_ 199	0.625996	1.154271	0.496933	1.080020	-0.836055
Cla_2	Ind_ 199	0.683523	1.186956	1.194800	1.516282	-0.200507
Cla_2	Ind_ 200	0.556810	1.334712	0.529374	0.184011	-1.378185
Cla_2	Ind_ 200	1.033784	1.413999	0.672597	1.102008	-1.004364
Cla_3	Ind_ 201	0.055559	0.966948	1.099963	-1.458052	0.547212
Cla_3	Ind_ 201	0.947188	1.510025	1.699652	-0.651418	1.106189
Cla_3	Ind_ 202	0.239448	0.876873	1.209204	-1.053529	0.261269
Cla_3	Ind_ 202	1.099734	0.895367	1.753319	-0.126774	1.145116
Cla_3	Ind_ 203	0.377819	0.967956	0.596896	-1.355037	1.292108
Cla_3	Ind_ 203	1.295969	1.186682	1.312250	-1.281060	1.521363
Cla_3	Ind_ 204	0.603778	1.307443	0.814493	-0.763985	0.870907
Cla_3	Ind_ 204	0.911863	1.444441	1.381771	-0.543886	1.448347
Cla_3	Ind_ 205	1.228965	0.397702	0.286157	-1.209113	0.543947
Cla_3	Ind_ 205	1.472381	0.703253	0.942671	-0.667887	1.442351
Cla_3	Ind_ 206	0.431516	0.825236	0.764031	-1.531602	0.879330
Cla_3	Ind_ 206	1.221458	1.412412	1.193396	-1.156011	1.338908
Cla_3	Ind_ 207	0.199881	0.993225	0.566286	-1.038743	0.191794
Cla_3	Ind_ 207	0.953749	1.071993	1.147359	-0.343043	0.818155
Cla_3	Ind_ 208	0.720115	0.876461	0.296564	-0.699316	0.774239
Cla_3	Ind_ 208	0.769829	1.777459	0.748956	-0.636509	1.441374
Cla_3	Ind_ 209	0.852107	0.358974	0.339991	-1.566012	1.112857
Cla_3	Ind_ 209	1.526414	0.859600	0.934919	-0.998856	1.403180
Cla_3	Ind_ 210	0.744011	0.345256	0.600757	-0.990982	0.749092
Cla_3	Ind_ 210	1.071078	0.683126	1.489303	-0.079455	1.367153
Cla_3	Ind_ 211	0.794626	0.785058	0.998627	-1.233711	0.642140
Cla_3	Ind_ 211	1.320338	0.873959	1.819605	-0.835231	1.313456

Cla_3	Ind_ 212	0.405438	0.836894	0.487228	-0.836543	0.899823
Cla_3	Ind_ 212	0.791772	1.232170	1.270913	-0.484756	1.725532
Cla_3	Ind_ 213	1.078234	0.420804	1.114063	-1.350780	0.390576
Cla_3	Ind_ 213	1.780221	0.779092	1.732429	-0.380810	1.112918
Cla_3	Ind_ 214	0.723121	0.699835	1.214286	-0.817530	0.340144
Cla_3	Ind_ 214	0.903424	0.731513	1.431028	-0.618580	0.750008
Cla_3	Ind_ 215	0.860500	0.883892	0.542314	-1.537172	0.365062
Cla_3	Ind_ 215	1.182562	1.623600	0.650807	-0.973983	1.039430
Cla_3	Ind_ 216	0.536592	0.490600	0.794610	-1.279748	1.314508
Cla_3	Ind_ 216	0.591403	1.249840	1.186010	-0.641301	1.526490
Cla_3	Ind_ 217	1.044191	1.131489	0.391003	-1.390530	0.593173
Cla_3	Ind_ 217	1.557970	1.516144	1.055116	-0.779305	0.729896
Cla_3	Ind_ 218	0.754662	1.333857	0.362011	-1.409375	0.135517
Cla_3	Ind_ 218	1.655477	1.595218	0.806207	-0.642048	1.054979
Cla_3	Ind_ 219	0.342433	0.672628	1.054247	-1.046587	0.205969
Cla_3	Ind_ 219	1.128040	1.433302	1.571291	-0.625797	1.053621
Cla_3	Ind_ 220	0.725639	0.381024	0.425459	-1.320276	1.025437
Cla_3	Ind_ 220	0.885403	1.073397	0.814966	-0.971328	1.131947
Cla_3	Ind_ 221	0.974059	0.949324	0.768288	-1.554613	1.182104
Cla_3	Ind_ 221	1.616443	1.393124	0.910138	-0.681448	1.413465
Cla_3	Ind_ 222	1.058168	0.821253	1.089267	-1.948546	1.188391
Cla_3	Ind_ 222	1.147252	1.135868	1.194830	-1.011963	1.785485
Cla_3	Ind_ 223	0.941481	0.964736	0.390500	-1.240516	1.205847
Cla_3	Ind_ 223	1.043107	0.979507	0.961318	-0.730888	1.624348
Cla_3	Ind_ 224	0.970077	0.646489	0.566851	-0.801798	1.192801
Cla_3	Ind_ 224	1.163259	1.276635	1.334483	-0.779458	1.248894
Cla_3	Ind_ 225	0.595202	0.986724	0.355327	-1.479034	0.842753
Cla_3	Ind_ 225	0.861660	1.451613	0.652181	-0.570864	1.728034
Cla_3	Ind_ 226	0.493728	1.125126	1.067110	-1.359905	0.991104
Cla_3	Ind_ 226	0.720878	1.787530	1.701743	-0.991760	1.206992
Cla_3	Ind_ 227	0.937162	0.904523	1.010041	-1.046846	0.985931
Cla_3	Ind_ 227	1.648479	0.922407	1.930692	-0.289590	1.353893
Cla_3	Ind_ 228	1.116016	1.186514	0.880306	-1.173345	1.155416
Cla_3	Ind_ 228	1.794870	1.731330	1.476241	-0.999420	1.798685
Cla_3	Ind_ 229	1.366710	1.066897	0.783380	-0.978820	0.775506
Cla_3	Ind_ 229	1.503464	1.841060	0.981536	-0.445570	0.985076
Cla_3	Ind_ 230	0.680960	0.669729	1.242897	-1.457701	1.157079
Cla_3	Ind_ 230	0.816492	0.741844	1.626545	-1.217032	1.611042
Cla_3	Ind_ 231	0.822092	1.180044	0.390667	-0.598804	0.945311
Cla_3	Ind_ 231	0.872082	1.263421	1.035371	-0.411267	1.739067
Cla_3	Ind_ 232	1.133320	0.640049	0.808374	-1.729148	0.268395
Cla_3	Ind_ 232	1.575289	1.011093	1.665853	-0.945769	1.149220
Cla_3	Ind_ 233	0.520386	1.092914	0.527726	-1.147465	1.299158
Cla_3	Ind_ 233	0.646977	1.144917	0.879696	-0.397778	1.522431
Cla_3	Ind_ 234	1.430738	0.725333	0.802942	-1.279473	0.914640
Cla_3	Ind_ 234	1.486007	1.071383	1.070986	-1.228690	1.015931
Cla_3	Ind_ 235	1.249489	1.048555	1.257607	-0.678213	0.270333
Cla_3	Ind_ 235	1.659932	1.140202	1.431898	-0.425489	1.169073
Cla_3	Ind_ 236	1.203543	0.676733	0.193777	-1.536592	0.553850
Cla_3	Ind_ 236	1.707282	0.733680	1.098773	-1.128437	0.785272
Cla_3	Ind_ 237	0.414457	0.729331	0.042467	-1.644078	0.767037
Cla_3	Ind_ 237	0.822184	1.644307	0.982437	-0.756661	0.920972
Cla_3	Ind_ 238	0.469196	1.105518	0.948119	-1.107593	0.413892
Cla_3	Ind_ 238	0.844188	1.789987	0.984039	-1.068712	1.305734
Cla_3	Ind_ 239	0.292627	0.674291	0.624454	-1.346324	0.257149
Cla_3	Ind_ 239	0.723853	1.558962	1.469146	-0.555864	0.861324
Cla_3	Ind_ 240	0.620441	1.100253	1.211219	-1.051531	0.663427
Cla_3	Ind_ 240	1.221137	1.339915	1.754234	-0.129688	0.915571
Cla_3	Ind_ 241	1.199286	0.896496	0.707846	-1.167867	1.295511
Cla_3	Ind_ 241	1.450545	1.019242	1.390271	-0.793222	1.659078
Cla_3	Ind_ 242	0.094607	0.676778	0.981567	-0.678274	0.845790
Cla_3	Ind_ 242	0.957518	1.350841	1.144078	-0.438307	1.806360
Cla_3	Ind_ 243	0.517518	0.778985	1.146718	-1.762246	0.446287
Cla_3	Ind_ 243	1.227317	1.064364	1.217521	-1.156575	1.234275
Cla_3	Ind_ 244	0.502777	0.264367	1.111209	-1.920927	0.653844
Cla_3	Ind_ 244	0.832194	0.810528	1.846065	-1.043672	1.517762
Cla_3	Ind_ 245	0.796106	0.576937	0.589877	-0.620579	0.304758

Cla_3	Ind_ 245	1.249336	0.987228	0.916547	-0.488067	1.204108
Cla_3	Ind_ 246	0.346324	0.181082	1.209250	-1.577929	0.373638
Cla_3	Ind_ 246	1.170110	1.082171	1.579012	-1.418653	1.299112
Cla_3	Ind_ 247	0.367168	0.604633	0.787729	-1.188803	0.569979
Cla_3	Ind_ 247	0.808191	1.347118	0.935926	-0.510926	0.717994
Cla_3	Ind_ 248	0.319651	0.301584	0.777032	-1.472854	1.074984
Cla_3	Ind_ 248	1.235725	0.820673	1.711722	-0.978057	1.320994
Cla_3	Ind_ 249	0.688009	0.434797	1.012711	-1.455809	0.872784
Cla_3	Ind_ 249	1.073550	1.379620	1.772988	-0.861492	1.542177
Cla_3	Ind_ 250	0.886517	0.870418	0.652745	-1.327280	0.877514
Cla_3	Ind_ 250	1.367672	1.060152	1.423582	-1.254006	1.472137
Cla_3	Ind_ 251	0.359279	1.189047	0.797433	-1.685888	0.311930
Cla_3	Ind_ 251	1.139515	1.788736	1.475982	-0.692938	1.002350
Cla_3	Ind_ 252	0.728553	0.663915	0.593677	-0.894787	0.369884
Cla_3	Ind_ 252	1.339228	0.831462	0.638264	-0.618900	0.725639
Cla_3	Ind_ 253	1.023438	1.247200	0.443419	-1.216117	0.400937
Cla_3	Ind_ 253	1.695059	1.654958	1.192709	-0.865825	0.644475
Cla_3	Ind_ 254	0.894192	0.453993	1.074633	-1.652745	0.510605
Cla_3	Ind_ 254	1.178411	0.919492	1.714301	-0.697455	0.808802
Cla_3	Ind_ 255	0.095767	0.953520	1.003159	-0.991287	0.379635
Cla_3	Ind_ 255	0.927488	1.943999	1.623783	-0.990982	1.222373
Cla_3	Ind_ 256	0.225517	0.976608	0.629612	-1.598270	0.588031
Cla_3	Ind_ 256	0.932569	1.119434	1.626164	-1.357295	0.660939
Cla_3	Ind_ 257	0.532105	1.148473	0.689199	-0.757866	0.592715
Cla_3	Ind_ 257	1.019425	1.850429	0.990875	-0.340770	0.871960
Cla_3	Ind_ 258	0.892010	0.858287	0.316401	-1.445585	1.068621
Cla_3	Ind_ 258	1.611881	1.101469	1.121906	-1.419340	1.914106
Cla_3	Ind_ 259	1.085177	0.603381	0.913099	-1.589129	0.889096
Cla_3	Ind_ 259	1.346995	0.960845	1.879192	-0.894131	1.081454
Cla_3	Ind_ 260	0.608325	0.704001	0.130696	-1.271905	0.910459
Cla_3	Ind_ 260	0.795312	1.243233	1.090136	-0.478912	1.879818
Cla_3	Ind_ 261	0.753731	1.186972	0.907163	-1.163121	0.244636
Cla_3	Ind_ 261	1.240318	1.470489	1.562670	-1.030885	0.816828
Cla_3	Ind_ 262	0.365337	0.799020	0.448912	-1.908185	1.171209
Cla_3	Ind_ 262	1.135044	1.539460	1.121357	-0.977493	1.278054
Cla_3	Ind_ 263	0.377239	0.340938	0.632572	-1.465575	0.561983
Cla_3	Ind_ 263	0.914579	0.938002	0.672735	-1.141163	1.411802
Cla_3	Ind_ 264	0.251564	0.628224	0.827219	-1.121693	0.487701
Cla_3	Ind_ 264	1.241554	0.948027	0.998306	-0.916059	1.191031
Cla_3	Ind_ 265	1.078829	0.693915	0.426603	-1.339290	0.349879
Cla_3	Ind_ 265	1.431227	0.812876	1.318720	-0.665663	1.082018
Cla_3	Ind_ 266	0.193854	1.166128	0.235206	-0.888089	1.256218
Cla_3	Ind_ 266	1.186041	1.782113	1.099094	-0.350871	1.477233
Cla_3	Ind_ 267	1.010407	0.390637	1.092364	-1.052324	1.235694
Cla_3	Ind_ 267	1.799158	1.219550	1.519868	-0.331599	1.672506
Cla_3	Ind_ 268	0.694815	0.640080	0.552553	-1.298547	0.590381
Cla_3	Ind_ 268	1.602466	1.007218	0.948012	-0.561953	0.629658
Cla_3	Ind_ 269	0.602359	0.465346	0.883770	-1.306345	0.514145
Cla_3	Ind_ 269	1.574679	0.732536	1.058336	-1.101840	0.997955
Cla_3	Ind_ 270	0.395764	0.720878	0.405896	-1.399762	0.886334
Cla_3	Ind_ 270	0.625294	1.053621	0.898068	-1.035096	1.313105
Cla_3	Ind_ 271	1.271432	0.837230	1.089694	-0.857479	0.685217
Cla_3	Ind_ 271	1.569933	1.627659	1.612629	-0.599811	1.385128
Cla_3	Ind_ 272	0.542787	0.842570	0.675222	-1.013581	0.571810
Cla_3	Ind_ 272	0.646123	1.001816	0.679647	-0.647816	1.064074
Cla_3	Ind_ 273	1.289682	1.255989	1.169134	-1.171285	1.184362
Cla_3	Ind_ 273	1.529160	1.342570	1.780084	-0.529115	1.776818
Cla_3	Ind_ 274	0.684362	1.105777	1.247063	-1.333018	0.868221
Cla_3	Ind_ 274	1.624271	1.252174	1.300104	-0.700522	1.388470
Cla_3	Ind_ 275	0.267632	0.275826	1.141911	-1.283776	0.258263
Cla_3	Ind_ 275	0.881970	1.132267	1.849178	-0.851146	1.006638
Cla_3	Ind_ 276	1.178426	0.966887	0.842814	-1.341029	0.643559
Cla_3	Ind_ 276	1.435087	1.037049	1.527772	-1.182272	1.241050
Cla_3	Ind_ 277	0.455077	0.585986	0.187048	-0.790384	0.689505
Cla_3	Ind_ 277	0.933363	0.652913	0.929533	-0.737220	1.104343
Cla_3	Ind_ 278	1.337214	0.666646	1.000656	-1.350078	0.184072
Cla_3	Ind_ 278	1.491394	1.385083	1.945875	-0.413190	1.120930

Cla_3	Ind_ 279	1.311884	0.275277	1.152364	-1.134617	0.322367
Cla_3	Ind_ 279	1.599368	0.958800	1.300317	-1.060915	1.240211
Cla_3	Ind_ 280	1.288110	0.601062	0.538591	-1.208213	0.463301
Cla_3	Ind_ 280	1.621067	0.896939	1.309702	-0.953261	1.268044
Cla_3	Ind_ 281	1.052873	0.360164	0.928266	-1.118671	1.094287
Cla_3	Ind_ 281	1.548067	0.646794	1.585208	-0.445982	1.169179
Cla_3	Ind_ 282	1.377850	1.087695	0.577410	-1.469207	1.235923
Cla_3	Ind_ 282	1.436110	1.442137	1.192287	-0.917982	1.694128
Cla_3	Ind_ 283	1.119465	0.653020	0.889004	-1.307245	1.024598
Cla_3	Ind_ 283	1.355495	1.132771	1.114139	-1.135548	1.282876
Cla_3	Ind_ 284	0.199822	1.230857	0.950896	-1.573412	0.535234
Cla_3	Ind_ 284	1.177526	1.635899	1.151921	-1.092532	1.533341
Cla_3	Ind_ 285	0.780755	0.799539	0.854411	-1.247108	1.365047
Cla_3	Ind_ 285	1.035188	1.294000	1.581240	-0.698172	1.629643
Cla_3	Ind_ 286	1.069765	0.838374	1.082263	-1.336268	0.902036
Cla_3	Ind_ 286	1.269753	1.705924	1.407407	-0.984024	1.446455
Cla_3	Ind_ 287	0.318201	0.563295	0.275277	-1.173742	0.863918
Cla_3	Ind_ 287	1.051378	1.462371	0.857418	-0.836604	1.511490
Cla_3	Ind_ 288	0.532807	0.593249	0.654347	-1.291131	0.422910
Cla_3	Ind_ 288	1.069689	1.108005	0.753502	-0.702673	1.063250
Cla_3	Ind_ 289	0.995972	0.185430	0.095462	-1.127232	0.348277
Cla_3	Ind_ 289	1.174902	1.002319	1.068239	-0.314432	1.017945
Cla_3	Ind_ 290	1.223273	1.433500	0.652058	-1.183767	0.406339
Cla_3	Ind_ 290	1.544359	1.500092	1.202948	-0.813272	0.697180
Cla_3	Ind_ 291	1.270899	0.897549	0.993851	-1.649342	0.575976
Cla_3	Ind_ 291	1.291177	1.458388	1.795267	-0.766961	1.105441
Cla_3	Ind_ 292	0.161824	1.342708	0.104480	-0.664754	0.274667
Cla_3	Ind_ 292	0.861675	1.434080	1.087481	-0.546434	0.817194
Cla_3	Ind_ 293	0.943724	0.916272	0.959426	-0.983215	1.061815
Cla_3	Ind_ 293	1.019745	1.006180	1.011185	-0.410901	1.082141
Cla_3	Ind_ 294	0.814341	0.505325	0.106189	-1.477340	0.999893
Cla_3	Ind_ 294	1.628758	1.374767	1.104083	-0.511093	1.010269
Cla_3	Ind_ 295	1.208792	0.497116	0.902402	-0.948256	1.330226
Cla_3	Ind_ 295	1.754952	0.691610	1.607654	-0.419675	1.607395
Cla_3	Ind_ 296	0.991256	1.029252	0.444060	-1.186850	0.661519
Cla_3	Ind_ 296	1.761727	1.405545	1.039293	-0.298700	0.920042
Cla_3	Ind_ 297	0.798242	0.376476	0.254585	-0.911420	0.798013
Cla_3	Ind_ 297	1.486404	1.192694	1.125004	-0.479156	1.565249
Cla_3	Ind_ 298	0.916578	1.188452	0.909406	-1.749611	0.954939
Cla_3	Ind_ 298	1.722266	1.229560	1.156972	-1.119404	1.395199
Cla_3	Ind_ 299	0.681143	0.678884	0.762230	-1.527161	0.739860
Cla_3	Ind_ 299	1.509568	1.087802	1.043886	-0.789468	0.921293
Cla_3	Ind_ 300	0.543458	0.389889	1.122272	-1.466857	1.227577
Cla_3	Ind_ 300	1.104328	0.826609	1.483245	-1.361141	1.332041
Cla_4	Ind_ 301	0.391659	0.644917	0.354808	-1.318400	-0.855052
Cla_4	Ind_ 301	0.611270	0.922361	1.074343	-0.601123	-0.483703
Cla_4	Ind_ 302	0.234280	0.300012	1.121921	-1.016266	-1.541810
Cla_4	Ind_ 302	1.095889	0.823466	1.195135	-0.527207	-0.940046
Cla_4	Ind_ 303	0.989975	0.319117	1.373821	-0.956114	-1.714011
Cla_4	Ind_ 303	1.380551	0.751930	1.426649	-0.379345	-0.862636
Cla_4	Ind_ 304	0.372829	0.340114	0.375774	-0.910001	-1.127094
Cla_4	Ind_ 304	1.325983	1.262017	0.902615	-0.272378	-0.274163
Cla_4	Ind_ 305	0.945296	0.302377	1.104205	-1.262673	-1.669530
Cla_4	Ind_ 305	1.705451	0.987793	1.274865	-0.455550	-1.082354
Cla_4	Ind_ 306	1.005188	0.653508	0.612781	-1.680776	-1.569628
Cla_4	Ind_ 306	1.741447	0.828196	1.079073	-1.226875	-0.674184
Cla_4	Ind_ 307	1.070269	0.837336	1.083178	-1.216971	-1.358730
Cla_4	Ind_ 307	1.485321	0.938688	1.326533	-1.098498	-0.797189
Cla_4	Ind_ 308	0.965896	0.370647	0.583651	-1.382046	-1.141835
Cla_4	Ind_ 308	1.338832	0.967254	1.015976	-1.205527	-0.915876
Cla_4	Ind_ 309	0.797800	0.787500	0.788263	-1.671056	-1.798013
Cla_4	Ind_ 309	1.761147	1.179754	1.616810	-0.761849	-1.020219
Cla_4	Ind_ 310	0.465896	0.611728	1.361202	-1.442122	-1.449141
Cla_4	Ind_ 310	1.130070	0.941939	1.480956	-0.682669	-0.717826
Cla_4	Ind_ 311	1.416517	0.881100	0.455840	-1.358791	-0.840724
Cla_4	Ind_ 311	1.518357	1.153783	0.794717	-0.632206	-0.181249
Cla_4	Ind_ 312	0.096240	0.643117	0.872082	-0.752602	-1.103977

Cla_4	Ind_ 312	0.991714	0.851650	1.261498	-0.524323	-0.956542
Cla_4	Ind_ 313	0.127216	1.346431	0.423795	-1.185797	-1.157949
Cla_4	Ind_ 313	1.125599	1.494283	0.904645	-0.721213	-1.081103
Cla_4	Ind_ 314	1.129170	1.214194	0.554704	-1.123707	-1.196203
Cla_4	Ind_ 314	1.454680	1.614597	0.883480	-0.647130	-0.722465
Cla_4	Ind_ 315	0.938766	1.079257	0.764489	-1.408338	-1.225364
Cla_4	Ind_ 315	1.205908	1.241768	1.709816	-0.579791	-0.443907
Cla_4	Ind_ 316	1.257866	1.340663	0.533418	-1.589679	-1.418302
Cla_4	Ind_ 316	1.608204	1.461455	1.426084	-1.307413	-1.044877
Cla_4	Ind_ 317	0.154042	1.233314	1.223365	-0.561525	-1.552019
Cla_4	Ind_ 317	0.987564	1.375866	1.353710	-0.462462	-1.021638
Cla_4	Ind_ 318	0.165731	1.049715	0.460270	-1.799402	-0.955550
Cla_4	Ind_ 318	1.026292	1.236152	0.605579	-0.972747	-0.863872
Cla_4	Ind_ 319	0.504379	0.374126	1.228568	-1.745949	-1.134098
Cla_4	Ind_ 319	0.931120	0.834437	1.410031	-1.019974	-0.453078
Cla_4	Ind_ 320	0.564714	0.852886	0.844462	-1.476272	-1.500214
Cla_4	Ind_ 320	1.429762	1.551515	1.549823	-0.980316	-0.577517
Cla_4	Ind_ 321	0.627644	1.087069	0.487945	-1.631809	-0.869167
Cla_4	Ind_ 321	0.679922	1.442091	0.611484	-0.848521	-0.331217
Cla_4	Ind_ 322	0.094333	0.961715	1.220084	-0.681967	-0.998795
Cla_4	Ind_ 322	0.936552	1.595828	1.433226	-0.434004	-0.242363
Cla_4	Ind_ 323	0.476692	0.179861	0.438215	-1.290750	-1.414350
Cla_4	Ind_ 323	0.630512	0.919385	0.810816	-0.473495	-1.181463
Cla_4	Ind_ 324	0.721747	0.512528	0.193350	-0.867977	-0.992599
Cla_4	Ind_ 324	1.641758	1.138127	0.899792	-0.544145	-0.255516
Cla_4	Ind_ 325	0.376057	1.370891	1.062258	-1.213553	-0.723426
Cla_4	Ind_ 325	1.170125	1.395581	1.467025	-1.095752	-0.304224
Cla_4	Ind_ 326	1.033219	0.315500	0.359401	-0.949721	-1.159948
Cla_4	Ind_ 326	1.407132	1.028260	0.937544	-0.362545	-0.595874
Cla_4	Ind_ 327	0.770104	0.102954	0.340541	-1.624165	-1.747032
Cla_4	Ind_ 327	1.378063	1.095538	1.205313	-0.678854	-1.185034
Cla_4	Ind_ 328	1.057970	0.393918	0.269158	-0.887738	-1.420209
Cla_4	Ind_ 328	1.546632	0.739204	1.134114	-0.616489	-0.953673
Cla_4	Ind_ 329	0.412442	0.874187	0.351802	-1.448805	-1.530061
Cla_4	Ind_ 329	1.237724	1.123768	1.246910	-0.705344	-1.424833
Cla_4	Ind_ 330	1.014817	0.828639	0.657735	-1.509735	-0.708533
Cla_4	Ind_ 330	1.326838	1.247871	1.481033	-1.094134	-0.325159
Cla_4	Ind_ 331	0.512589	0.725623	0.788537	-1.049959	-1.191366
Cla_4	Ind_ 331	0.656484	1.016770	1.755821	-0.179907	-0.894177
Cla_4	Ind_ 332	1.122196	1.174551	1.313776	-1.058397	-0.781243
Cla_4	Ind_ 332	1.455000	1.802133	1.649602	-0.284997	-0.401990
Cla_4	Ind_ 333	0.618748	0.777093	0.403500	-1.584078	-1.019150
Cla_4	Ind_ 333	1.339778	1.485946	1.399625	-0.707770	-0.064104
Cla_4	Ind_ 334	0.362560	0.463225	0.731620	-1.199210	-1.160192
Cla_4	Ind_ 334	0.976257	1.389991	0.910611	-0.728492	-0.331828
Cla_4	Ind_ 335	1.238945	0.451353	1.152318	-1.092761	-1.667425
Cla_4	Ind_ 335	1.316584	1.022721	1.171026	-0.226463	-0.717017
Cla_4	Ind_ 336	0.341182	1.058794	0.436110	-0.972488	-1.576861
Cla_4	Ind_ 336	0.751778	1.318751	1.119633	-0.773324	-1.179968
Cla_4	Ind_ 337	0.472030	0.277367	1.047914	-1.187063	-1.686926
Cla_4	Ind_ 337	0.663411	0.787881	1.268823	-0.219138	-0.990127
Cla_4	Ind_ 338	0.630528	0.881268	0.620640	-0.924177	-1.024461
Cla_4	Ind_ 338	1.439940	1.620060	1.438841	-0.701575	-0.668615
Cla_4	Ind_ 339	0.663076	0.913282	0.992889	-1.433287	-0.875866
Cla_4	Ind_ 339	1.223243	1.457060	1.571612	-0.987106	-0.455626
Cla_4	Ind_ 340	1.240501	0.764321	0.989593	-1.399792	-1.277841
Cla_4	Ind_ 340	1.652257	1.628422	1.218940	-0.419691	-1.023103
Cla_4	Ind_ 341	0.348521	1.279870	0.604434	-0.800989	-1.684194
Cla_4	Ind_ 341	0.866268	1.443937	1.542512	-0.310465	-1.189825
Cla_4	Ind_ 342	1.105243	0.462004	0.445387	-1.235649	-1.490692
Cla_4	Ind_ 342	1.705908	0.729957	0.711722	-0.670385	-1.002335
Cla_4	Ind_ 343	1.007447	0.444746	0.798685	-0.781457	-1.655614
Cla_4	Ind_ 343	1.589892	1.173437	1.542665	-0.574450	-1.034440
Cla_4	Ind_ 344	1.227409	0.570818	0.511032	-1.332163	-1.728996
Cla_4	Ind_ 344	1.645146	1.407971	1.035432	-0.467910	-0.942137
Cla_4	Ind_ 345	0.883724	1.148473	0.648900	-1.036470	-1.142750
Cla_4	Ind_ 345	1.004059	1.826624	0.919233	-0.189367	-1.004013

Cla_4	Ind_ 346	0.870052	0.882183	0.128346	-1.343211	-1.050737
Cla_4	Ind_ 346	1.452742	1.022263	1.065203	-0.710227	-0.889294
Cla_4	Ind_ 347	0.895322	1.112537	0.284219	-1.687063	-1.278542
Cla_4	Ind_ 347	0.999603	1.728828	0.874050	-0.883297	-1.041261
Cla_4	Ind_ 348	0.436949	0.431379	0.223029	-1.466018	-1.490722
Cla_4	Ind_ 348	1.123859	0.849574	1.169927	-1.170934	-1.154653
Cla_4	Ind_ 349	0.898191	1.020569	1.343654	-1.100665	-1.439009
Cla_4	Ind_ 349	1.145833	1.803766	1.422452	-0.694098	-1.074648
Cla_4	Ind_ 350	0.932340	0.758477	0.176977	-1.566240	-1.559847
Cla_4	Ind_ 350	1.024293	1.681387	0.968047	-0.824366	-1.363186
Cla_4	Ind_ 351	0.806024	0.812632	0.251518	-0.787576	-1.681982
Cla_4	Ind_ 351	1.114414	1.764321	0.988296	-0.740761	-1.080767
Cla_4	Ind_ 352	0.639927	0.646443	0.598209	-1.622166	-1.118671
Cla_4	Ind_ 352	1.204520	1.526078	1.023209	-0.706732	-0.829234
Cla_4	Ind_ 353	0.874142	0.683981	0.830012	-1.264183	-1.395047
Cla_4	Ind_ 353	1.676809	0.963073	0.931608	-0.600223	-1.104511
Cla_4	Ind_ 354	0.412748	0.852748	1.029511	-1.722678	-1.234397
Cla_4	Ind_ 354	0.591769	1.712790	1.152898	-1.266457	-0.335322
Cla_4	Ind_ 355	0.592486	0.928358	0.513855	-1.494049	-1.638691
Cla_4	Ind_ 355	1.571490	1.073565	0.555483	-1.369411	-0.940641
Cla_4	Ind_ 356	1.213614	0.599780	0.986129	-0.603336	-0.880276
Cla_4	Ind_ 356	1.732002	1.443403	1.270684	-0.406980	-0.142064
Cla_4	Ind_ 357	1.024018	0.516511	0.683523	-0.765297	-1.917386
Cla_4	Ind_ 357	1.288461	1.024400	0.889492	-0.239708	-0.929472
Cla_4	Ind_ 358	0.297250	0.339396	0.963057	-0.955016	-1.587649
Cla_4	Ind_ 358	1.003296	1.089389	1.434751	-0.175115	-0.807840
Cla_4	Ind_ 359	0.561296	0.551103	0.882092	-1.113819	-1.364254
Cla_4	Ind_ 359	0.567309	1.358592	1.296289	-0.162374	-1.069933
Cla_4	Ind_ 360	0.674978	1.057741	1.302881	-1.489090	-0.909085
Cla_4	Ind_ 360	0.973174	1.781121	1.524445	-1.297952	-0.646168
Cla_4	Ind_ 361	0.381909	0.987762	0.931440	-1.151860	-1.573046
Cla_4	Ind_ 361	1.156377	1.939116	1.698431	-0.432752	-1.403272
Cla_4	Ind_ 362	0.251610	0.635640	1.064364	-0.994110	-1.302728
Cla_4	Ind_ 362	0.821024	1.585589	1.716453	-0.521409	-0.898785
Cla_4	Ind_ 363	0.615329	1.263604	0.213599	-0.911206	-1.061922
Cla_4	Ind_ 363	0.629185	1.673223	0.972015	-0.271538	-0.870174
Cla_4	Ind_ 364	1.276513	0.530290	1.282952	-1.368053	-1.296625
Cla_4	Ind_ 364	1.497955	0.820063	1.682867	-0.954070	-0.452849
Cla_4	Ind_ 365	0.987487	0.223563	1.132771	-1.060472	-1.103458
Cla_4	Ind_ 365	1.305521	1.177267	1.304407	-0.731666	-1.094882
Cla_4	Ind_ 366	0.998199	0.793985	0.763070	-0.860317	-1.543092
Cla_4	Ind_ 366	1.045778	1.298639	1.682470	-0.368450	-1.180105
Cla_4	Ind_ 367	0.458708	0.485397	0.523499	-0.605609	-0.905179
Cla_4	Ind_ 367	0.574007	0.553728	0.562441	-0.507736	-0.298868
Cla_4	Ind_ 368	1.164144	1.195029	0.793710	-1.508194	-1.621647
Cla_4	Ind_ 368	1.747353	1.557588	1.467803	-0.663961	-0.726112
Cla_4	Ind_ 369	1.120151	0.633229	1.020966	-0.933378	-0.875347
Cla_4	Ind_ 369	1.162786	1.202429	1.468581	-0.477706	-0.798074
Cla_4	Ind_ 370	0.755531	0.457320	0.427244	-1.002106	-1.200735
Cla_4	Ind_ 370	1.029160	0.857814	0.679388	-0.443709	-0.828104
Cla_4	Ind_ 371	0.864605	0.881420	0.658528	-1.806955	-0.957549
Cla_4	Ind_ 371	1.655095	1.337611	1.131108	-1.053636	-0.285501
Cla_4	Ind_ 372	0.598285	0.383297	0.212378	-1.463881	-0.916959
Cla_4	Ind_ 372	0.789605	1.010147	1.149144	-1.408979	-0.489059
Cla_4	Ind_ 373	0.597842	0.598544	1.208869	-1.172750	-0.769585
Cla_4	Ind_ 373	1.297327	1.133473	1.226173	-1.170186	-0.353893
Cla_4	Ind_ 374	0.240379	0.376507	1.021867	-1.372356	-1.581637
Cla_4	Ind_ 374	1.073168	1.320780	1.784707	-0.523850	-1.086322
Cla_4	Ind_ 375	0.505020	0.832316	0.533311	-0.888699	-1.530015
Cla_4	Ind_ 375	1.395703	1.668950	1.377789	-0.489181	-1.460097
Cla_4	Ind_ 376	1.236793	1.129231	0.865459	-1.121021	-1.073717
Cla_4	Ind_ 376	1.263558	1.703772	1.456908	-0.654027	-0.743141
Cla_4	Ind_ 377	1.138676	0.726157	0.456328	-1.232307	-1.507370
Cla_4	Ind_ 377	1.348430	1.094424	1.033006	-0.647908	-0.785333
Cla_4	Ind_ 378	0.731910	0.990524	1.329173	-1.441862	-1.528825
Cla_4	Ind_ 378	1.217215	1.257500	1.562120	-0.631840	-1.040803
Cla_4	Ind_ 379	0.343623	0.467910	0.429319	-0.798563	-0.976012

Cla_4	Ind_ 379	1.110981	0.692068	1.379116	-0.629551	-0.194037
Cla_4	Ind_ 380	1.480773	0.280908	1.030625	-1.679479	-1.183203
Cla_4	Ind_ 380	1.509217	0.851604	1.180837	-0.951430	-0.399762
Cla_4	Ind_ 381	0.784402	0.661672	0.500504	-1.646962	-0.913678
Cla_4	Ind_ 381	1.140492	1.251442	0.599536	-1.083743	-0.244469
Cla_4	Ind_ 382	0.698904	0.568865	0.732124	-0.650624	-1.482864
Cla_4	Ind_ 382	0.712912	1.521928	1.505402	-0.546892	-1.208106
Cla_4	Ind_ 383	1.090732	0.262764	0.856014	-1.806543	-1.923917
Cla_4	Ind_ 383	1.616718	0.932432	1.015900	-1.175512	-0.954009
Cla_4	Ind_ 384	1.382702	0.674429	0.501287	-1.072817	-1.816126
Cla_4	Ind_ 384	1.386395	0.719321	1.457625	-0.856594	-1.024567
Cla_4	Ind_ 385	0.386929	0.966979	0.947996	-0.999268	-1.618870
Cla_4	Ind_ 385	0.986007	1.037507	1.912229	-0.790735	-1.272607
Cla_4	Ind_ 386	0.489090	0.916776	1.060121	-1.796594	-0.980010
Cla_4	Ind_ 386	0.686026	1.710776	1.669057	-1.126621	-0.435652
Cla_4	Ind_ 387	1.220786	0.194464	0.974868	-0.690115	-1.946638
Cla_4	Ind_ 387	1.625767	0.864803	1.088763	-0.676107	-0.967483
Cla_4	Ind_ 388	1.117283	0.369854	1.213477	-1.325983	-1.239341
Cla_4	Ind_ 388	1.780541	1.177709	1.245308	-0.656713	-0.393735
Cla_4	Ind_ 389	0.657140	1.059297	0.730293	-1.323969	-1.196753
Cla_4	Ind_ 389	0.675512	1.564470	0.942518	-0.341700	-0.371929
Cla_4	Ind_ 390	0.081103	1.142842	0.496918	-1.861583	-1.087725
Cla_4	Ind_ 390	0.986160	1.810221	1.486511	-0.941816	-0.614689
Cla_4	Ind_ 391	1.052614	0.464278	1.085498	-1.357891	-1.169408
Cla_4	Ind_ 391	1.332194	1.169286	1.620548	-0.804437	-0.701590
Cla_4	Ind_ 392	1.014466	0.450407	0.935286	-1.850017	-1.570406
Cla_4	Ind_ 392	1.795892	0.942854	0.985580	-0.911817	-0.863536
Cla_4	Ind_ 393	1.276193	1.213385	0.621220	-1.949599	-1.741432
Cla_4	Ind_ 393	1.655751	1.222633	1.543550	-1.000595	-0.891003
Cla_4	Ind_ 394	0.398953	0.988922	0.551744	-1.275140	-1.777749
Cla_4	Ind_ 394	0.723731	1.529618	0.776330	-1.223746	-1.205588
Cla_4	Ind_ 395	0.891903	0.780374	1.174673	-1.909589	-1.814676
Cla_4	Ind_ 395	1.182714	1.674505	1.346675	-0.945204	-1.154134
Cla_4	Ind_ 396	1.155339	1.383633	1.018967	-1.674551	-0.869060
Cla_4	Ind_ 396	1.636586	1.504181	1.453368	-0.808161	-0.551668
Cla_4	Ind_ 397	1.069277	0.126011	0.817042	-0.777917	-1.380398
Cla_4	Ind_ 397	1.394330	0.968047	1.533830	-0.466720	-0.721442
Cla_4	Ind_ 398	1.188986	0.222877	0.092135	-1.615833	-1.236396
Cla_4	Ind_ 398	1.494903	0.823267	0.935331	-1.123966	-0.575640
Cla_4	Ind_ 399	0.236290	0.203467	0.527512	-0.634312	-1.184210
Cla_4	Ind_ 399	0.835551	0.988617	1.392987	-0.595187	-0.456862
Cla_4	Ind_ 400	0.530091	0.445097	0.715125	-1.617710	-1.666753
Cla_4	Ind_ 400	1.174886	1.239250	1.293817	-1.324244	-0.831584

Fichier "artificielles.sds"

```

SODAS = (
CONTAINS = (
  FILES, HEADER, INDIVIDUALS, VARIABLES, RECTANGLE_MATRIX
),
FILE = (
  procedure_name = "db2ao" ,
  version = "sans" ,
  create_date = ""
),
HEADER = (
  title = "Données artificielles" ,
  indiv_nb = 400 ,
  var_nb = 5 ,
  rules_nb = 0 ,
  nb_var_set = 0 ,
  nb_indiv_set = 0 ,
  nb_var_nom = 0 ,
  nb_var_cont = 0 ,
  nb_var_text = 0 ,
  nb_var_cont_symb = 5 ,
  nb_var_nom_symb = 0 ,

```

```

nb_var_nom_mod = 0 ,
nb_na = 0 ,
nb_null = 0 ,
nb_nu = 0 ,
nb_hierarchies = 0
),

INDIVIDUALS = (
(0,"i0", "Ind_ 1" ),
(1,"i1", "Ind_ 2" ),
(2,"i2", "Ind_ 3" ),
(3,"i3", "Ind_ 4" ),
(4,"i4", "Ind_ 5" ),
(5,"i5", "Ind_ 6" ),
(6,"i6", "Ind_ 7" ),
(7,"i7", "Ind_ 8" ),
(8,"i8", "Ind_ 9" ),
(9,"i9", "Ind_ 10" ),
(10,"i10", "Ind_ 11" ),
(11,"i11", "Ind_ 12" ),
(12,"i12", "Ind_ 13" ),
(13,"i13", "Ind_ 14" ),
(14,"i14", "Ind_ 15" ),
(15,"i15", "Ind_ 16" ),
(16,"i16", "Ind_ 17" ),
(17,"i17", "Ind_ 18" ),
(18,"i18", "Ind_ 19" ),
(19,"i19", "Ind_ 20" ),
(20,"i20", "Ind_ 21" ),
(21,"i21", "Ind_ 22" ),
(22,"i22", "Ind_ 23" ),
(23,"i23", "Ind_ 24" ),
(24,"i24", "Ind_ 25" ),
(25,"i25", "Ind_ 26" ),
(26,"i26", "Ind_ 27" ),
(27,"i27", "Ind_ 28" ),
(28,"i28", "Ind_ 29" ),
(29,"i29", "Ind_ 30" ),
(30,"i30", "Ind_ 31" ),
(31,"i31", "Ind_ 32" ),
(32,"i32", "Ind_ 33" ),
(33,"i33", "Ind_ 34" ),
(34,"i34", "Ind_ 35" ),
(35,"i35", "Ind_ 36" ),
(36,"i36", "Ind_ 37" ),
(37,"i37", "Ind_ 38" ),
(38,"i38", "Ind_ 39" ),
(39,"i39", "Ind_ 40" ),
(40,"i40", "Ind_ 41" ),
(41,"i41", "Ind_ 42" ),
(42,"i42", "Ind_ 43" ),
(43,"i43", "Ind_ 44" ),
(44,"i44", "Ind_ 45" ),
(45,"i45", "Ind_ 46" ),
(46,"i46", "Ind_ 47" ),
(47,"i47", "Ind_ 48" ),
(48,"i48", "Ind_ 49" ),
(49,"i49", "Ind_ 50" ),
(50,"i50", "Ind_ 51" ),
(51,"i51", "Ind_ 52" ),
(52,"i52", "Ind_ 53" ),
(53,"i53", "Ind_ 54" ),
(54,"i54", "Ind_ 55" ),
(55,"i55", "Ind_ 56" ),
(56,"i56", "Ind_ 57" ),
(57,"i57", "Ind_ 58" ),
(58,"i58", "Ind_ 59" ),

```

(59,"i59", "Ind_ 60"),
(60,"i60", "Ind_ 61"),
(61,"i61", "Ind_ 62"),
(62,"i62", "Ind_ 63"),
(63,"i63", "Ind_ 64"),
(64,"i64", "Ind_ 65"),
(65,"i65", "Ind_ 66"),
(66,"i66", "Ind_ 67"),
(67,"i67", "Ind_ 68"),
(68,"i68", "Ind_ 69"),
(69,"i69", "Ind_ 70"),
(70,"i70", "Ind_ 71"),
(71,"i71", "Ind_ 72"),
(72,"i72", "Ind_ 73"),
(73,"i73", "Ind_ 74"),
(74,"i74", "Ind_ 75"),
(75,"i75", "Ind_ 76"),
(76,"i76", "Ind_ 77"),
(77,"i77", "Ind_ 78"),
(78,"i78", "Ind_ 79"),
(79,"i79", "Ind_ 80"),
(80,"i80", "Ind_ 81"),
(81,"i81", "Ind_ 82"),
(82,"i82", "Ind_ 83"),
(83,"i83", "Ind_ 84"),
(84,"i84", "Ind_ 85"),
(85,"i85", "Ind_ 86"),
(86,"i86", "Ind_ 87"),
(87,"i87", "Ind_ 88"),
(88,"i88", "Ind_ 89"),
(89,"i89", "Ind_ 90"),
(90,"i90", "Ind_ 91"),
(91,"i91", "Ind_ 92"),
(92,"i92", "Ind_ 93"),
(93,"i93", "Ind_ 94"),
(94,"i94", "Ind_ 95"),
(95,"i95", "Ind_ 96"),
(96,"i96", "Ind_ 97"),
(97,"i97", "Ind_ 98"),
(98,"i98", "Ind_ 99"),
(99,"i99", "Ind_ 100"),
(100,"i100", "Ind_ 101"),
(101,"i101", "Ind_ 102"),
(102,"i102", "Ind_ 103"),
(103,"i103", "Ind_ 104"),
(104,"i104", "Ind_ 105"),
(105,"i105", "Ind_ 106"),
(106,"i106", "Ind_ 107"),
(107,"i107", "Ind_ 108"),
(108,"i108", "Ind_ 109"),
(109,"i109", "Ind_ 110"),
(110,"i110", "Ind_ 111"),
(111,"i111", "Ind_ 112"),
(112,"i112", "Ind_ 113"),
(113,"i113", "Ind_ 114"),
(114,"i114", "Ind_ 115"),
(115,"i115", "Ind_ 116"),
(116,"i116", "Ind_ 117"),
(117,"i117", "Ind_ 118"),
(118,"i118", "Ind_ 119"),
(119,"i119", "Ind_ 120"),
(120,"i120", "Ind_ 121"),
(121,"i121", "Ind_ 122"),
(122,"i122", "Ind_ 123"),
(123,"i123", "Ind_ 124"),
(124,"i124", "Ind_ 125"),
(125,"i125", "Ind_ 126"),

(126,"i126", "Ind_ 127"),
(127,"i127", "Ind_ 128"),
(128,"i128", "Ind_ 129"),
(129,"i129", "Ind_ 130"),
(130,"i130", "Ind_ 131"),
(131,"i131", "Ind_ 132"),
(132,"i132", "Ind_ 133"),
(133,"i133", "Ind_ 134"),
(134,"i134", "Ind_ 135"),
(135,"i135", "Ind_ 136"),
(136,"i136", "Ind_ 137"),
(137,"i137", "Ind_ 138"),
(138,"i138", "Ind_ 139"),
(139,"i139", "Ind_ 140"),
(140,"i140", "Ind_ 141"),
(141,"i141", "Ind_ 142"),
(142,"i142", "Ind_ 143"),
(143,"i143", "Ind_ 144"),
(144,"i144", "Ind_ 145"),
(145,"i145", "Ind_ 146"),
(146,"i146", "Ind_ 147"),
(147,"i147", "Ind_ 148"),
(148,"i148", "Ind_ 149"),
(149,"i149", "Ind_ 150"),
(150,"i150", "Ind_ 151"),
(151,"i151", "Ind_ 152"),
(152,"i152", "Ind_ 153"),
(153,"i153", "Ind_ 154"),
(154,"i154", "Ind_ 155"),
(155,"i155", "Ind_ 156"),
(156,"i156", "Ind_ 157"),
(157,"i157", "Ind_ 158"),
(158,"i158", "Ind_ 159"),
(159,"i159", "Ind_ 160"),
(160,"i160", "Ind_ 161"),
(161,"i161", "Ind_ 162"),
(162,"i162", "Ind_ 163"),
(163,"i163", "Ind_ 164"),
(164,"i164", "Ind_ 165"),
(165,"i165", "Ind_ 166"),
(166,"i166", "Ind_ 167"),
(167,"i167", "Ind_ 168"),
(168,"i168", "Ind_ 169"),
(169,"i169", "Ind_ 170"),
(170,"i170", "Ind_ 171"),
(171,"i171", "Ind_ 172"),
(172,"i172", "Ind_ 173"),
(173,"i173", "Ind_ 174"),
(174,"i174", "Ind_ 175"),
(175,"i175", "Ind_ 176"),
(176,"i176", "Ind_ 177"),
(177,"i177", "Ind_ 178"),
(178,"i178", "Ind_ 179"),
(179,"i179", "Ind_ 180"),
(180,"i180", "Ind_ 181"),
(181,"i181", "Ind_ 182"),
(182,"i182", "Ind_ 183"),
(183,"i183", "Ind_ 184"),
(184,"i184", "Ind_ 185"),
(185,"i185", "Ind_ 186"),
(186,"i186", "Ind_ 187"),
(187,"i187", "Ind_ 188"),
(188,"i188", "Ind_ 189"),
(189,"i189", "Ind_ 190"),
(190,"i190", "Ind_ 191"),
(191,"i191", "Ind_ 192"),
(192,"i192", "Ind_ 193"),

(193,"i193", "Ind_ 194"),
(194,"i194", "Ind_ 195"),
(195,"i195", "Ind_ 196"),
(196,"i196", "Ind_ 197"),
(197,"i197", "Ind_ 198"),
(198,"i198", "Ind_ 199"),
(199,"i199", "Ind_ 200"),
(200,"i200", "Ind_ 201"),
(201,"i201", "Ind_ 202"),
(202,"i202", "Ind_ 203"),
(203,"i203", "Ind_ 204"),
(204,"i204", "Ind_ 205"),
(205,"i205", "Ind_ 206"),
(206,"i206", "Ind_ 207"),
(207,"i207", "Ind_ 208"),
(208,"i208", "Ind_ 209"),
(209,"i209", "Ind_ 210"),
(210,"i210", "Ind_ 211"),
(211,"i211", "Ind_ 212"),
(212,"i212", "Ind_ 213"),
(213,"i213", "Ind_ 214"),
(214,"i214", "Ind_ 215"),
(215,"i215", "Ind_ 216"),
(216,"i216", "Ind_ 217"),
(217,"i217", "Ind_ 218"),
(218,"i218", "Ind_ 219"),
(219,"i219", "Ind_ 220"),
(220,"i220", "Ind_ 221"),
(221,"i221", "Ind_ 222"),
(222,"i222", "Ind_ 223"),
(223,"i223", "Ind_ 224"),
(224,"i224", "Ind_ 225"),
(225,"i225", "Ind_ 226"),
(226,"i226", "Ind_ 227"),
(227,"i227", "Ind_ 228"),
(228,"i228", "Ind_ 229"),
(229,"i229", "Ind_ 230"),
(230,"i230", "Ind_ 231"),
(231,"i231", "Ind_ 232"),
(232,"i232", "Ind_ 233"),
(233,"i233", "Ind_ 234"),
(234,"i234", "Ind_ 235"),
(235,"i235", "Ind_ 236"),
(236,"i236", "Ind_ 237"),
(237,"i237", "Ind_ 238"),
(238,"i238", "Ind_ 239"),
(239,"i239", "Ind_ 240"),
(240,"i240", "Ind_ 241"),
(241,"i241", "Ind_ 242"),
(242,"i242", "Ind_ 243"),
(243,"i243", "Ind_ 244"),
(244,"i244", "Ind_ 245"),
(245,"i245", "Ind_ 246"),
(246,"i246", "Ind_ 247"),
(247,"i247", "Ind_ 248"),
(248,"i248", "Ind_ 249"),
(249,"i249", "Ind_ 250"),
(250,"i250", "Ind_ 251"),
(251,"i251", "Ind_ 252"),
(252,"i252", "Ind_ 253"),
(253,"i253", "Ind_ 254"),
(254,"i254", "Ind_ 255"),
(255,"i255", "Ind_ 256"),
(256,"i256", "Ind_ 257"),
(257,"i257", "Ind_ 258"),
(258,"i258", "Ind_ 259"),
(259,"i259", "Ind_ 260"),

(260,"i260", "Ind_ 261"),
(261,"i261", "Ind_ 262"),
(262,"i262", "Ind_ 263"),
(263,"i263", "Ind_ 264"),
(264,"i264", "Ind_ 265"),
(265,"i265", "Ind_ 266"),
(266,"i266", "Ind_ 267"),
(267,"i267", "Ind_ 268"),
(268,"i268", "Ind_ 269"),
(269,"i269", "Ind_ 270"),
(270,"i270", "Ind_ 271"),
(271,"i271", "Ind_ 272"),
(272,"i272", "Ind_ 273"),
(273,"i273", "Ind_ 274"),
(274,"i274", "Ind_ 275"),
(275,"i275", "Ind_ 276"),
(276,"i276", "Ind_ 277"),
(277,"i277", "Ind_ 278"),
(278,"i278", "Ind_ 279"),
(279,"i279", "Ind_ 280"),
(280,"i280", "Ind_ 281"),
(281,"i281", "Ind_ 282"),
(282,"i282", "Ind_ 283"),
(283,"i283", "Ind_ 284"),
(284,"i284", "Ind_ 285"),
(285,"i285", "Ind_ 286"),
(286,"i286", "Ind_ 287"),
(287,"i287", "Ind_ 288"),
(288,"i288", "Ind_ 289"),
(289,"i289", "Ind_ 290"),
(290,"i290", "Ind_ 291"),
(291,"i291", "Ind_ 292"),
(292,"i292", "Ind_ 293"),
(293,"i293", "Ind_ 294"),
(294,"i294", "Ind_ 295"),
(295,"i295", "Ind_ 296"),
(296,"i296", "Ind_ 297"),
(297,"i297", "Ind_ 298"),
(298,"i298", "Ind_ 299"),
(299,"i299", "Ind_ 300"),
(300,"i300", "Ind_ 301"),
(301,"i301", "Ind_ 302"),
(302,"i302", "Ind_ 303"),
(303,"i303", "Ind_ 304"),
(304,"i304", "Ind_ 305"),
(305,"i305", "Ind_ 306"),
(306,"i306", "Ind_ 307"),
(307,"i307", "Ind_ 308"),
(308,"i308", "Ind_ 309"),
(309,"i309", "Ind_ 310"),
(310,"i310", "Ind_ 311"),
(311,"i311", "Ind_ 312"),
(312,"i312", "Ind_ 313"),
(313,"i313", "Ind_ 314"),
(314,"i314", "Ind_ 315"),
(315,"i315", "Ind_ 316"),
(316,"i316", "Ind_ 317"),
(317,"i317", "Ind_ 318"),
(318,"i318", "Ind_ 319"),
(319,"i319", "Ind_ 320"),
(320,"i320", "Ind_ 321"),
(321,"i321", "Ind_ 322"),
(322,"i322", "Ind_ 323"),
(323,"i323", "Ind_ 324"),
(324,"i324", "Ind_ 325"),
(325,"i325", "Ind_ 326"),
(326,"i326", "Ind_ 327"),

(327,"i327", "Ind_ 328"),
(328,"i328", "Ind_ 329"),
(329,"i329", "Ind_ 330"),
(330,"i330", "Ind_ 331"),
(331,"i331", "Ind_ 332"),
(332,"i332", "Ind_ 333"),
(333,"i333", "Ind_ 334"),
(334,"i334", "Ind_ 335"),
(335,"i335", "Ind_ 336"),
(336,"i336", "Ind_ 337"),
(337,"i337", "Ind_ 338"),
(338,"i338", "Ind_ 339"),
(339,"i339", "Ind_ 340"),
(340,"i340", "Ind_ 341"),
(341,"i341", "Ind_ 342"),
(342,"i342", "Ind_ 343"),
(343,"i343", "Ind_ 344"),
(344,"i344", "Ind_ 345"),
(345,"i345", "Ind_ 346"),
(346,"i346", "Ind_ 347"),
(347,"i347", "Ind_ 348"),
(348,"i348", "Ind_ 349"),
(349,"i349", "Ind_ 350"),
(350,"i350", "Ind_ 351"),
(351,"i351", "Ind_ 352"),
(352,"i352", "Ind_ 353"),
(353,"i353", "Ind_ 354"),
(354,"i354", "Ind_ 355"),
(355,"i355", "Ind_ 356"),
(356,"i356", "Ind_ 357"),
(357,"i357", "Ind_ 358"),
(358,"i358", "Ind_ 359"),
(359,"i359", "Ind_ 360"),
(360,"i360", "Ind_ 361"),
(361,"i361", "Ind_ 362"),
(362,"i362", "Ind_ 363"),
(363,"i363", "Ind_ 364"),
(364,"i364", "Ind_ 365"),
(365,"i365", "Ind_ 366"),
(366,"i366", "Ind_ 367"),
(367,"i367", "Ind_ 368"),
(368,"i368", "Ind_ 369"),
(369,"i369", "Ind_ 370"),
(370,"i370", "Ind_ 371"),
(371,"i371", "Ind_ 372"),
(372,"i372", "Ind_ 373"),
(373,"i373", "Ind_ 374"),
(374,"i374", "Ind_ 375"),
(375,"i375", "Ind_ 376"),
(376,"i376", "Ind_ 377"),
(377,"i377", "Ind_ 378"),
(378,"i378", "Ind_ 379"),
(379,"i379", "Ind_ 380"),
(380,"i380", "Ind_ 381"),
(381,"i381", "Ind_ 382"),
(382,"i382", "Ind_ 383"),
(383,"i383", "Ind_ 384"),
(384,"i384", "Ind_ 385"),
(385,"i385", "Ind_ 386"),
(386,"i386", "Ind_ 387"),
(387,"i387", "Ind_ 388"),
(388,"i388", "Ind_ 389"),
(389,"i389", "Ind_ 390"),
(390,"i390", "Ind_ 391"),
(391,"i391", "Ind_ 392"),
(392,"i392", "Ind_ 393"),
(393,"i393", "Ind_ 394"),

```

(394,"i394", "Ind_ 395" ),
(395,"i395", "Ind_ 396" ),
(396,"i396", "Ind_ 397" ),
(397,"i397", "Ind_ 398" ),
(398,"i398", "Ind_ 399" ),
(399,"i399", "Ind_ 400" )
),
VARIABLES = (
(1 ,inter_cont ,"" , "v1" ,"Variable_n°1" ,0, 0, 0.055559, 1.93963),
(2 ,inter_cont ,"" , "v2" ,"Variable_n°2" ,0, 0, 0.071017, 1.99664),
(3 ,inter_cont ,"" , "v3" ,"Variable_n°3" ,0, 0, 0.033448, 1.94588),
(4 ,inter_cont ,"" , "v4" ,"Variable_n°4" ,0, 0, -1.9496, 1.9405),
(5 ,inter_cont ,"" , "v5" ,"Variable_n°5" ,0, 0, -1.96332, 1.96941)
),
RECTANGLE_MATRIX = (
(( 0.383374 : 0.618946 ), ( 1.11097 : 1.18534 ), ( 0.590228 : 0.950255 ), ( 0.262642 : 1.24793 ),
( 0.351802 : 1.28604 )),
(( 0.24984 : 1.10741 ), ( 0.530564 : 0.53914 ), ( 0.883816 : 1.3697 ), ( 0.371899 : 1.17408 ),
( 0.367458 : 0.81547 )),
(( 0.653523 : 1.62679 ), ( 0.765572 : 1.55618 ), ( 1.04309 : 1.08347 ), ( 1.34722 : 1.41559 ),
( 0.304254 : 1.28933 )),
(( 0.565645 : 0.583651 ), ( 0.242012 : 0.809259 ), ( 0.691916 : 1.34004 ), ( 1.02805 : 1.4131 ),
( 0.866909 : 0.893368 )),
(( 1.29635 : 1.50626 ), ( 1.08838 : 1.63427 ), ( 1.1502 : 1.77839 ), ( 0.397961 : 1.34742 ),
( 0.470855 : 0.991684 )),
(( 0.284143 : 1.18151 ), ( 0.668661 : 1.08081 ), ( 1.06653 : 1.42204 ), ( 0.766762 : 0.985763 ),
( 0.277184 : 0.865093 )),
(( 0.959181 : 1.39059 ), ( 1.30499 : 1.33862 ), ( 1.03255 : 1.28903 ), ( 0.817667 : 1.66779 ),
( 0.29284 : 1.04885 )),
(( 0.728599 : 1.13184 ), ( 0.388043 : 0.897794 ), ( 0.722449 : 1.20897 ), ( 0.166112 : 1.00156 ),
( 1.10637 : 1.65583 )),
(( 0.960738 : 1.05693 ), ( 0.502045 : 1.21935 ), ( 0.443419 : 0.571169 ), ( 0.144017 : 1.02713 ),
( 0.434553 : 0.749962 )),
(( 0.800195 : 1.45308 ), ( 0.812296 : 1.16552 ), ( 1.07993 : 1.506 ), ( 0.411237 : 0.62685 ),
( 1.14559 : 1.53714 )),
(( 1.05083 : 1.42425 ), ( 1.09301 : 1.81434 ), ( 0.584094 : 1.36924 ), ( 0.516434 : 1.25874 ),
( 0.607883 : 1.30291 )),
(( 0.523804 : 0.891919 ), ( 0.224403 : 0.993194 ), ( 0.217566 : 0.789697 ), ( 0.995621 : 1.59439 ),
( 0.907422 : 1.08956 )),
(( 0.980224 : 1.08939 ), ( 0.569597 : 1.34877 ), ( 0.494568 : 1.38791 ), ( 0.819315 : 1.8 ),
( 1.48769 : 1.50096 )),
(( 0.4111 : 1.30102 ), ( 1.17119 : 1.46545 ), ( 0.033448 : 1.02612 ), ( 0.294794 : 0.938551 ),
( 0.4129 : 0.695013 )),
(( 0.707938 : 1.13584 ), ( 0.993149 : 1.88722 ), ( 1.098 : 1.53328 ), ( 1.20525 : 1.23766 ),
( 0.545991 : 1.00621 )),
(( 0.594089 : 1.06188 ), ( 0.790582 : 0.826685 ), ( 0.817011 : 1.20591 ), ( 0.840098 : 1.19088 ),
( 0.280099 : 1.1692 )),
(( 0.324412 : 1.1988 ), ( 0.927839 : 1.7967 ), ( 0.22631 : 0.957961 ), ( 0.451476 : 1.08693 ),
( 0.383541 : 0.733039 )),
(( 0.932447 : 1.9138 ), ( 0.902036 : 1.31327 ), ( 0.685446 : 0.707389 ), ( 0.370617 : 1.01929 ),
( 0.317713 : 0.892682 )),
(( 0.51854 : 1.02188 ), ( 0.769616 : 1.13098 ), ( 1.2056 : 1.39381 ), ( 0.852123 : 1.66257 ),
( 1.38618 : 1.56706 )),
(( 1.04746 : 1.79672 ), ( 0.437239 : 1.11716 ), ( 1.08751 : 1.84577 ), ( 0.991791 : 1.04669 ),
( 1.07669 : 1.55086 )),
(( 0.16686 : 0.989395 ), ( 0.673803 : 1.29516 ), ( 1.32891 : 1.50745 ), ( 1.09986 : 1.15274 ),
( 0.784631 : 1.40284 )),
(( 0.985839 : 1.56865 ), ( 1.04032 : 1.71602 ), ( 0.327616 : 1.21806 ), ( 0.518387 : 0.832148 ),
( 0.562716 : 0.841105 )),
(( 0.373074 : 1.36834 ), ( 0.534471 : 1.09977 ), ( 0.295053 : 1.26234 ), ( 0.359981 : 1.25436 ),
( 0.44615 : 1.23594 )),
(( 0.46675 : 1.20631 ), ( 0.914426 : 1.5537 ), ( 0.313868 : 1.20385 ), ( 0.830485 : 1.24993 ),
( 0.563585 : 1.25773 )),
(( 0.535447 : 0.744774 ), ( 0.718436 : 1.06369 ), ( 0.941511 : 1.02251 ), ( 0.990005 : 1.47885 ),
( 0.475143 : 1.29252 )),
(( 0.914777 : 1.84846 ), ( 0.744316 : 1.30601 ), ( 0.482528 : 1.23014 ), ( 0.319803 : 1.24992 ),
( 0.793207 : 1.31278 )),

```

((0.271233 : 0.889294), (0.507614 : 0.576128), (0.848994 : 0.881375), (1.1077 : 1.48885),
 (0.951796 : 1.93175)),
 ((0.732093 : 0.791299), (0.771645 : 1.71546), (1.10257 : 1.16047), (0.755455 : 1.73476),
 (0.306574 : 1.30331)),
 ((0.27517 : 1.03496), (0.95233 : 1.77569), (0.646519 : 1.4508), (0.961104 : 1.07396),
 (0.386608 : 0.804804)),
 ((0.513321 : 1.29591), (0.851924 : 1.22984), (0.913221 : 1.41757), (0.146413 : 1.03725),
 (0.888318 : 1.48349)),
 ((0.941069 : 1.80325), (0.58887 : 1.25674), (0.094867 : 1.09447), (0.30343 : 1.12597),
 (0.718604 : 1.50543)),
 ((1.15552 : 1.81585), (0.071017 : 0.929044), (0.297586 : 1.1984), (0.929411 : 1.73275),
 (0.867733 : 1.7517)),
 ((0.323756 : 1.26147), (0.724509 : 1.46284), (1.27621 : 1.33993), (0.7257 : 1.22214),
 (0.935804 : 1.69553)),
 ((0.567019 : 1.50552), (0.749535 : 0.974181), (0.945585 : 1.39174), (1.06004 : 1.27093),
 (1.06427 : 1.71148)),
 ((0.770562 : 0.846644), (0.742882 : 0.875179), (0.468245 : 0.953642), (0.650884 : 0.676519),
 (0.145817 : 0.941221)),
 ((0.928205 : 1.61823), (0.412595 : 1.06356), (1.20913 : 1.5571), (0.335185 : 1.16135),
 (0.315271 : 1.21581)),
 ((0.299249 : 1.13973), (1.13324 : 1.85711), (1.03166 : 1.41369), (0.441939 : 1.16831),
 (1.17495 : 1.24319)),
 ((0.706503 : 1.15271), (0.553957 : 1.12984), (0.603046 : 1.07453), (1.02556 : 1.69703),
 (0.790643 : 1.75445)),
 ((0.551775 : 0.725089), (0.696249 : 1.2627), (0.387616 : 0.764183), (0.327616 : 0.806604),
 (0.52324 : 1.49529)),
 ((0.380383 : 1.35896), (0.772103 : 1.76292), (1.15445 : 1.32472), (0.644948 : 1.45024),
 (0.54593 : 0.792886)),
 ((0.400723 : 1.11907), (1.12972 : 1.16982), (0.914686 : 1.08583), (1.09841 : 1.24136),
 (1.2548 : 1.35676)),
 ((0.573412 : 0.960723), (0.606357 : 1.09258), (0.55388 : 1.50883), (0.93646 : 1.52586),
 (1.01869 : 1.92369)),
 ((0.537645 : 1.50218), (0.947417 : 1.3462), (0.842692 : 1.56726), (0.776238 : 1.17924),
 (1.15014 : 1.27172)),
 ((1.05355 : 1.20632), (0.151524 : 1.1157), (0.811533 : 0.967238), (0.901685 : 1.80491),
 (0.919523 : 0.937559)),
 ((1.1209 : 1.23992), (0.933927 : 1.80856), (0.471786 : 0.9111), (0.211188 : 0.914548),
 (1.19987 : 1.62038)),
 ((0.798944 : 1.10489), (0.778985 : 1.202), (0.871029 : 1.10666), (0.452589 : 0.664541),
 (1.22243 : 1.67594)),
 ((0.810907 : 0.925321), (0.588855 : 1.58373), (1.29475 : 1.55968), (0.807718 : 1.23382),
 (0.766976 : 1.73655)),
 ((0.585055 : 0.959395), (1.04199 : 1.76287), (0.885006 : 1.79528), (0.513413 : 1.4077),
 (1.20641 : 1.27227)),
 ((0.271645 : 0.928983), (0.763283 : 1.03136), (1.35084 : 1.47398), (0.833979 : 1.80383),
 (0.996445 : 1.27581)),
 ((1.29487 : 1.55443), (1.09426 : 1.81773), (0.604205 : 1.33461), (0.150655 : 0.886822),
 (1.25221 : 1.64275)),
 ((0.965835 : 1.68979), (0.441847 : 1.33753), (0.889279 : 1.45134), (0.517533 : 1.40272),
 (0.299722 : 1.16862)),
 ((0.905759 : 1.60863), (0.698904 : 1.4423), (1.08469 : 1.36842), (0.559984 : 1.55681),
 (0.347072 : 0.835429)),
 ((0.322001 : 0.731529), (0.373501 : 1.03862), (0.670202 : 0.71482), (0.380001 : 1.10299),
 (0.993011 : 1.14252)),
 ((0.857021 : 1.35163), (1.15651 : 1.83293), (0.664815 : 1.33699), (0.987106 : 1.9405),
 (0.500626 : 0.597095)),
 ((1.10241 : 1.63959), (1.00383 : 1.68482), (0.481689 : 1.26028), (0.514222 : 1.26589),
 (0.916318 : 1.04843)),
 ((0.749397 : 0.83375), (0.730232 : 1.47153), (0.916593 : 1.33705), (0.411939 : 0.742119),
 (0.524689 : 1.26708)),
 ((0.548357 : 1.4814), (0.912366 : 1.00246), (0.700522 : 1.19367), (1.257 : 1.45711),
 (0.612918 : 1.47482)),
 ((1.08148 : 1.82897), (0.396039 : 0.625568), (0.523118 : 0.996246), (0.57329 : 1.09226),
 (0.381115 : 1.35792)),
 ((0.132862 : 0.906323), (1.208 : 1.66541), (0.975112 : 1.04494), (0.460265 : 0.642338),
 (0.895749 : 1.88427)),
 ((0.975036 : 1.93963), (0.996536 : 1.16744), (0.786462 : 1.22156), (0.66242 : 1.65915),

(0.549547 : 0.931669),
 ((0.997375 : 1.90826), (0.585376 : 1.53325), (0.858623 : 1.29113), (0.814386 : 1.77288),
 (0.380062 : 0.757027)),
 ((1.09279 : 1.26302), (1.1258 : 1.44725), (0.218085 : 0.931211), (0.763756 : 1.61037),
 (0.619007 : 1.13169)),
 ((0.401044 : 0.985382), (0.495682 : 1.46761), (0.470489 : 1.07414), (0.925764 : 1.61969),
 (0.8099 : 0.975768)),
 ((1.2018 : 1.72886), (0.824107 : 1.20394), (0.415754 : 1.12925), (0.871029 : 1.65996),
 (1.05643 : 1.47883)),
 ((1.26151 : 1.68264), (0.634632 : 1.08414), (1.0143 : 1.34732), (0.603076 : 0.966582),
 (1.25654 : 1.43577)),
 ((0.585238 : 1.46655), (0.910413 : 1.749), (0.179571 : 0.932432), (1.05631 : 1.68429),
 (0.354778 : 1.29185)),
 ((1.11376 : 1.24938), (1.20118 : 1.55113), (0.729987 : 0.934584), (0.554247 : 1.52068),
 (0.517426 : 0.615268)),
 ((0.256172 : 0.812738), (0.530229 : 0.906552), (0.570269 : 1.40852), (1.12079 : 1.64208),
 (1.01447 : 1.34123)),
 ((0.566774 : 1.39847), (0.629627 : 1.29282), (1.10942 : 1.73792), (0.447752 : 1.14998),
 (0.793115 : 1.08997)),
 ((0.676214 : 0.817576), (1.24799 : 1.44524), (0.36317 : 1.3354), (0.305002 : 1.28224),
 (1.08052 : 1.5956)),
 ((0.585055 : 0.89964), (1.24883 : 1.70771), (0.951979 : 1.57566), (0.541398 : 0.807794),
 (0.970779 : 1.72642)),
 ((0.931837 : 1.85923), (1.26109 : 1.48512), (0.600925 : 0.740699), (0.886959 : 1.87445),
 (0.549364 : 1.17039)),
 ((1.17328 : 1.49278), (1.07082 : 1.2607), (0.589633 : 1.38788), (0.331568 : 1.25292),
 (0.092196 : 1.03818)),
 ((1.07233 : 1.22642), (0.564211 : 1.11495), (0.69747 : 1.33094), (0.745643 : 0.94615),
 (0.767953 : 1.3381)),
 ((0.792917 : 0.854564), (0.162908 : 1.05896), (0.35902 : 0.791803), (0.926023 : 1.42491),
 (0.83491 : 0.940138)),
 ((1.06166 : 1.77818), (1.1422 : 1.2036), (0.797494 : 1.66947), (0.71894 : 1.26125),
 (0.578127 : 0.709357)),
 ((0.587878 : 0.603229), (0.805658 : 1.26994), (1.14846 : 1.22256), (0.093463 : 1.08422),
 (0.583178 : 1.38365)),
 ((1.04784 : 1.53974), (0.518952 : 1.14025), (0.343623 : 0.713324), (1.33349 : 1.52405),
 (0.109867 : 0.944334)),
 ((0.748314 : 1.07605), (0.389996 : 0.992218), (0.598453 : 1.14278), (0.693335 : 1.57642),
 (0.419187 : 0.608707)),
 ((0.902753 : 1.84474), (0.741295 : 1.48756), (0.554613 : 0.5674), (1.23434 : 1.54312),
 (0.756325 : 1.6037)),
 ((0.589785 : 1.36312), (0.257118 : 0.937468), (0.283776 : 1.05644), (0.937925 : 1.82952),
 (0.110218 : 0.996536)),
 ((0.735679 : 0.882443), (0.625309 : 0.782479), (1.26801 : 1.56597), (0.393689 : 0.8399),
 (0.673421 : 1.49855)),
 ((0.676885 : 1.33303), (1.43348 : 1.53337), (0.954955 : 1.31059), (1.19134 : 1.44433),
 (0.921583 : 1.75669)),
 ((0.833567 : 1.7598), (0.843669 : 0.968917), (0.850108 : 1.54056), (0.925428 : 0.93556),
 (0.737907 : 1.14908)),
 ((1.06839 : 1.63356), (0.964675 : 1.30697), (0.667272 : 0.919233), (0.757195 : 0.986908),
 (1.33279 : 1.43405)),
 ((1.23113 : 1.43719), (1.18154 : 1.19619), (0.62653 : 1.38412), (1.0844 : 1.80122),
 (0.666158 : 1.2107)),
 ((1.01309 : 1.40266), (0.800974 : 1.15743), (0.376171 : 0.894772), (0.859081 : 1.84245),
 (0.985 : 1.96941)),
 ((0.363842 : 0.648), (0.540071 : 1.46863), (0.536195 : 0.769967), (0.778588 : 1.20545),
 (0.747459 : 0.977233)),
 ((0.673574 : 1.17362), (0.937773 : 1.37806), (0.825861 : 1.31968), (0.266152 : 0.740013),
 (0.438383 : 0.797708)),
 ((1.00755 : 1.16076), (1.01482 : 1.87989), (0.462279 : 1.13576), (0.671834 : 0.943785),
 (0.894559 : 1.82107)),
 ((1.20682 : 1.70907), (0.764092 : 1.03659), (1.31062 : 1.64477), (0.267769 : 0.807978),
 (1.06677 : 1.71954)),
 ((0.456603 : 0.993301), (0.627232 : 0.916425), (0.442946 : 1.29127), (0.70571 : 1.18439),
 (0.964873 : 1.17872)),
 ((0.778771 : 1.24775), (1.32202 : 1.57324), (0.635838 : 0.946516), (0.449507 : 1.26847),
 (0.668218 : 1.0284)),

((0.1451 : 0.938704), (1.03818 : 1.56508), (1.21511 : 1.75956), (0.118305 : 0.914167),
 (0.555086 : 1.44195)),
 ((1.34251 : 1.39937), (0.930692 : 1.72591), (0.85786 : 1.62961), (0.474303 : 1.13125),
 (1.12001 : 1.63047)),
 ((1.16317 : 1.51199), (0.32049 : 1.20827), (0.294275 : 1.21117), (0.788461 : 1.72013),
 (0.722755 : 0.863506)),
 ((0.756157 : 1.17051), (0.877483 : 1.64203), (0.629673 : 0.807352), (0.821558 : 1.49599),
 (0.154805 : 1.13866)),
 ((0.871838 : 1.67704), (0.119541 : 1.04221), (1.04387 : 1.10021), (0.982818 : 1.88974),
 (0.837642 : 1.83691)),
 ((1.07086 : 1.35752), (1.12967 : 1.78616), (0.731651 : 0.845698), (0.687201 : 0.822031),
 (1.08802 : 1.17765)),
 ((0.570193 : 0.963271), (0.779504 : 1.50136), (0.436506 : 1.36204), (0.33488 : 1.06516),
 (1.01613 : 1.65928)),
 ((0.219459 : 0.783044), (0.288934 : 1.09767), (0.845073 : 1.32495), (0.40231 : 1.29827),
 (-1.05046 : -0.303858)),
 ((0.244636 : 1.10358), (0.953734 : 1.46727), (0.796503 : 0.811487), (0.409177 : 0.773629),
 (-1.43564 : -1.26974)),
 ((1.26568 : 1.71137), (0.616749 : 0.621418), (0.319971 : 0.697851), (0.746071 : 1.31726),
 (-1.20182 : -0.594653)),
 ((0.334712 : 0.997757), (0.774728 : 1.12685), (0.253197 : 0.860881), (0.882015 : 1.68462),
 (-1.13109 : -0.829142)),
 ((1.01264 : 1.73931), (0.993042 : 1.91876), (0.968184 : 1.11052), (0.844417 : 1.07975),
 (-0.742561 : -0.53296)),
 ((0.85783 : 1.70148), (0.996948 : 1.99664), (0.915281 : 1.30772), (0.617573 : 0.914853),
 (-0.671728 : -0.647984)),
 ((0.829554 : 0.922178), (1.1491 : 1.20531), (0.049394 : 0.968184), (0.639439 : 0.912336),
 (-1.25768 : -0.5665)),
 ((0.974364 : 1.70086), (0.88226 : 1.08762), (1.00951 : 1.47797), (0.483383 : 1.43254),
 (-0.809702 : -0.701422)),
 ((0.90643 : 1.29167), (0.930525 : 1.53949), (0.891736 : 1.25307), (0.539003 : 0.764107),
 (-1.47629 : -0.673406)),
 ((0.522111 : 1.5121), (1.07877 : 1.42433), (0.340327 : 0.997635), (0.960128 : 1.02367),
 (-1.05264 : -0.547838)),
 ((0.172704 : 1.12229), (0.189016 : 1.09413), (1.04137 : 1.34442), (0.891369 : 0.961745),
 (-0.874981 : -0.191794)),
 ((0.214606 : 1.09186), (1.03066 : 1.61271), (0.602405 : 0.780297), (1.07956 : 1.55483),
 (-1.59641 : -1.09249)),
 ((1.02922 : 1.43481), (0.495209 : 1.06395), (0.804315 : 1.56017), (0.984268 : 1.45956),
 (-1.56088 : -1.19308)),
 ((1.31713 : 1.35223), (0.685522 : 1.34851), (0.873882 : 0.978561), (0.988647 : 1.91003),
 (-1.12345 : -0.777459)),
 ((0.784234 : 1.15922), (1.18854 : 1.50542), (0.820154 : 1.09204), (1.33407 : 1.63187),
 (-1.04445 : -0.477172)),
 ((0.315332 : 1.07665), (1.14061 : 1.53827), (0.555818 : 1.44598), (0.030152 : 1.02478),
 (-0.952666 : -0.902158)),
 ((0.934294 : 1.12836), (1.02966 : 1.65642), (1.05869 : 1.25654), (1.2805 : 1.40382),
 (-1.76164 : -1.01851)),
 ((0.343532 : 1.2846), (0.617924 : 0.954237), (0.273721 : 1.00681), (0.980621 : 1.68862),
 (-1.27337 : -0.526154)),
 ((0.680486 : 0.824961), (0.471114 : 0.532121), (0.879925 : 1.73255), (0.652776 : 0.76838),
 (-0.953917 : -0.939665)),
 ((0.386517 : 0.841029), (0.909146 : 1.59529), (1.0065 : 1.08039), (0.835749 : 1.03769),
 (-0.948958 : -0.658605)),
 ((0.820475 : 1.0529), (0.811579 : 1.34416), (1.04859 : 1.20878), (0.522614 : 1.48566),
 (-1.26664 : -0.341838)),
 ((0.521973 : 0.857921), (0.180761 : 1.17594), (0.458434 : 1.45645), (0.284921 : 0.910092),
 (-1.62447 : -1.18674)),
 ((1.4073 : 1.45573), (1.24961 : 1.53963), (0.342769 : 1.11183), (0.809732 : 1.0117),
 (-1.174 : -0.569857)),
 ((0.718436 : 1.18479), (0.780465 : 1.41519), (0.940397 : 1.76919), (0.764321 : 1.48523),
 (-1.12182 : -0.746681)),
 ((0.315317 : 1.05322), (0.602588 : 1.50768), (0.648396 : 0.837336), (0.75547 : 1.45398),
 (-1.09104 : -0.739738)),
 ((0.954268 : 1.03465), (0.934721 : 1.54677), (0.774819 : 1.46594), (1.22997 : 1.37909),
 (-1.35783 : -0.490097)),
 ((1.10421 : 1.71891), (1.20608 : 1.24929), (0.67951 : 1.65604), (0.530412 : 1.09961),

```

( -1.28114 : -1.10721 ),
(( 0.174032 : 1.04308 ), ( 0.979064 : 1.72338 ), ( 0.491424 : 0.818339 ), ( 0.541047 : 0.617649 ),
( -1.26902 : -0.449019 ),
(( 0.820963 : 1.26922 ), ( 0.759606 : 1.05835 ), ( 0.714957 : 1.21616 ), ( 0.491134 : 0.814173 ),
( -0.918943 : -0.60506 ),
(( 0.847148 : 1.80622 ), ( 1.01083 : 1.73586 ), ( 0.328059 : 1.27206 ), ( 0.594363 : 0.6601 ),
( -0.977325 : -0.452742 ),
(( 0.631681 : 1.5877 ), ( 0.134449 : 1.01009 ), ( 0.992798 : 1.31492 ), ( 0.352275 : 0.857326 ),
( -1.41806 : -1.12777 ),
(( 1.1444 : 1.69556 ), ( 1.10553 : 1.22007 ), ( 0.802973 : 1.1821 ), ( 0.600131 : 1.39349 ),
( -1.18192 : -0.799554 ),
(( 0.922086 : 1.45424 ), ( 0.908689 : 1.30387 ), ( 0.151952 : 0.859828 ), ( 0.289087 : 0.912152 ),
( -0.882504 : -0.391003 ),
(( 0.998886 : 1.49579 ), ( 0.487426 : 1.27279 ), ( 0.874264 : 1.23136 ), ( 1.14029 : 1.77114 ),
( -1.51054 : -1.1363 ),
(( 0.259987 : 1.00327 ), ( 1.14573 : 1.75771 ), ( 0.362911 : 0.692755 ), ( 0.236305 : 0.875515 ),
( -1.79191 : -0.944838 ),
(( 1.06587 : 1.66276 ), ( 0.794656 : 1.64863 ), ( 0.451445 : 0.577914 ), ( 0.899335 : 1.51648 ),
( -1.31541 : -1.24946 ),
(( 0.356868 : 0.980972 ), ( 0.68128 : 1.00069 ), ( 0.537065 : 1.19806 ), ( 0.898953 : 1.70583 ),
( -1.27902 : -0.667913 ),
(( 0.84788 : 1.74848 ), ( 0.329722 : 0.959899 ), ( 0.775567 : 1.02927 ), ( 0.208945 : 1.06414 ),
( -1.64774 : -1.21993 ),
(( 0.922208 : 1.22449 ), ( 0.93527 : 1.16083 ), ( 0.756035 : 0.866665 ), ( 1.24068 : 1.37539 ),
( -1.60981 : -0.821696 ),
(( 1.00041 : 1.79005 ), ( 0.936186 : 1.55141 ), ( 0.432798 : 1.28945 ), ( 0.296701 : 1.16028 ),
( -1.39534 : -1.14579 ),
(( 0.549989 : 1.53482 ), ( 0.513092 : 0.594516 ), ( 0.811274 : 1.23808 ), ( 0.46527 : 0.724067 ),
( -0.724845 : -0.49208 ),
(( 0.584002 : 0.709098 ), ( 1.39158 : 1.47169 ), ( 0.517731 : 0.576449 ), ( 0.379055 : 1.29376 ),
( -1.31779 : -0.885006 ),
(( 1.02757 : 1.86476 ), ( 0.613178 : 1.45528 ), ( 0.99469 : 1.39238 ), ( 0.756996 : 0.76133 ),
( -1.45181 : -0.497009 ),
(( 0.778146 : 1.01924 ), ( 0.957991 : 1.21313 ), ( 0.711371 : 1.65665 ), ( 0.490387 : 1.38061 ),
( -1.96332 : -1.02234 ),
(( 0.708472 : 1.49463 ), ( 1.00546 : 1.14789 ), ( 0.530824 : 0.913831 ), ( 0.295312 : 0.713233 ),
( -1.74772 : -1.08779 ),
(( 1.32267 : 1.38752 ), ( 0.980026 : 1.64209 ), ( 0.790139 : 1.59284 ), ( 0.687323 : 1.37295 ),
( -1.70199 : -1.01248 ),
(( 0.83903 : 1.61676 ), ( 0.096728 : 0.965407 ), ( 0.791238 : 1.4978 ), ( 0.309458 : 0.861446 ),
( -0.581484 : -0.522706 ),
(( 0.702383 : 0.84756 ), ( 1.17179 : 1.79177 ), ( 0.331004 : 1.25349 ), ( 0.520264 : 1.2148 ),
( -1.35932 : -1.20341 ),
(( 0.479827 : 1.00127 ), ( 1.34907 : 1.45549 ), ( 1.18178 : 1.62351 ), ( 0.189062 : 0.97116 ),
( -1.81541 : -0.841014 ),
(( 0.840678 : 1.71107 ), ( 0.48233 : 0.938948 ), ( 0.128437 : 0.87907 ), ( 0.411695 : 0.816401 ),
( -1.68517 : -0.692557 ),
(( 0.412534 : 0.664617 ), ( 0.565432 : 0.81312 ), ( 0.343348 : 0.963668 ), ( 0.91583 : 1.85492 ),
( -1.69417 : -0.914518 ),
(( 0.817179 : 1.47394 ), ( 1.18142 : 1.63688 ), ( 1.08921 : 1.75338 ), ( 0.332591 : 0.968932 ),
( -1.14148 : -0.719443 ),
(( 1.17274 : 1.71331 ), ( 0.810343 : 1.3468 ), ( 0.55681 : 0.953764 ), ( 0.83195 : 0.868572 ),
( -0.803003 : -0.606494 ),
(( 0.375576 : 0.764992 ), ( 1.05528 : 1.12621 ), ( 0.619724 : 0.775613 ), ( 0.917173 : 1.3715 ),
( -1.24438 : -0.547029 ),
(( 0.599063 : 1.28353 ), ( 0.478668 : 1.31439 ), ( 0.992416 : 1.19201 ), ( 1.01146 : 1.88746 ),
( -1.60202 : -0.614566 ),
(( 0.237449 : 1.13317 ), ( 0.85342 : 1.29548 ), ( 0.77308 : 1.4816 ), ( 0.406842 : 0.692404 ),
( -1.44362 : -1.03598 ),
(( 1.04022 : 1.75042 ), ( 0.780419 : 1.67647 ), ( 0.694861 : 1.09227 ), ( 1.24957 : 1.55817 ),
( -1.39718 : -0.826609 ),
(( 0.486709 : 1.22041 ), ( 0.877163 : 1.61354 ), ( 1.16977 : 1.30909 ), ( 0.563875 : 0.836467 ),
( -1.27512 : -0.363994 ),
(( 0.609012 : 1.12667 ), ( 0.15566 : 1.0631 ), ( 0.44409 : 0.96205 ), ( 0.935575 : 1.3741 ),
( -0.85757 : -0.767144 ),
(( 0.541673 : 0.61745 ), ( 0.350261 : 0.705344 ), ( 0.458586 : 1.15587 ), ( 0.44525 : 0.840098 ),
( -1.77003 : -1.09429 ),

```


((1.12558 : 1.32429), (0.886166 : 1.50188), (0.8829 : 1.42689), (0.325617 : 0.870983),
 (-1.93767 : -0.963088)),
 ((0.479049 : 1.44902), (0.962386 : 1.49335), (0.833522 : 1.52632), (0.677801 : 1.06642),
 (-1.85493 : -1.04515)),
 ((0.639149 : 0.699973), (0.418317 : 1.2028), (0.551515 : 0.873424), (0.980087 : 0.989242),
 (-1.31407 : -0.806207)),
 ((0.726676 : 1.48521), (0.726173 : 1.07598), (0.871487 : 1.81381), (0.624668 : 1.62195),
 (-1.72796 : -1.16511)),
 ((1.17017 : 1.55693), (0.664296 : 0.948515), (0.22042 : 0.832987), (0.881878 : 0.900494),
 (-1.69681 : -1.24111)),
 ((0.889554 : 1.82632), (0.538255 : 1.26554), (0.707404 : 1.51578), (1.08412 : 1.66672),
 (-0.944487 : -0.865474)),
 ((1.08208 : 1.52428), (0.795907 : 1.38534), (0.661168 : 1.47723), (1.00096 : 1.34454),
 (-1.62343 : -1.35612)),
 ((0.725517 : 0.95761), (1.08515 : 1.45689), (1.31144 : 1.4283), (0.911115 : 1.54004),
 (-0.697562 : -0.514542)),
 ((0.584567 : 1.48657), (0.797479 : 1.76269), (0.24659 : 1.14814), (0.820719 : 1.77412),
 (-1.09636 : -0.206015)),
 ((1.03598 : 1.22144), (0.984436 : 1.24943), (0.384548 : 1.11675), (1.20264 : 1.37358),
 (-1.02658 : -0.54564)),
 ((0.178213 : 0.984756), (1.29302 : 1.41766), (0.520814 : 1.09413), (0.591952 : 1.354),
 (-1.67202 : -0.944243)),
 ((1.48735 : 1.5027), (0.801813 : 1.49345), (0.638081 : 0.710196), (0.436659 : 1.11329),
 (-0.939405 : -0.382748)),
 ((0.481475 : 1.26098), (0.413404 : 1.05164), (0.629368 : 0.833964), (1.33016 : 1.61191),
 (-1.86172 : -1.0925)),
 ((0.628346 : 0.675314), (0.334956 : 0.683538), (0.972259 : 1.3155), (0.706305 : 1.12282),
 (-1.68192 : -1.13674)),
 ((1.03027 : 1.36827), (0.719398 : 1.59565), (0.750465 : 1.32049), (0.3211 : 0.786874),
 (-1.45341 : -1.02661)),
 ((1.12005 : 1.60663), (0.304529 : 0.870342), (0.82017 : 1.59864), (0.938948 : 1.7895),
 (-1.63918 : -0.774804)),
 ((0.771493 : 1.51834), (0.94583 : 1.71145), (0.917753 : 1.82607), (1.15156 : 1.67351),
 (-1.11365 : -1.00601)),
 ((0.442488 : 1.26554), (0.842799 : 1.01264), (0.172124 : 0.829585), (1.24169 : 1.30541),
 (-1.60302 : -1.21735)),
 ((0.780084 : 1.5877), (0.582659 : 0.725517), (0.375393 : 0.721534), (0.029344 : 1.00777),
 (-0.879192 : -0.825541)),
 ((1.09542 : 1.73289), (0.921186 : 0.942244), (0.546617 : 0.938566), (0.76307 : 1.54495),
 (-1.89019 : -1.06415)),
 ((0.321299 : 0.955565), (0.558657 : 1.54215), (0.856029 : 1.07082), (0.409619 : 1.19178),
 (-1.39776 : -0.917158)),
 ((1.05676 : 1.15928), (1.10653 : 1.86235), (0.422681 : 0.879086), (0.712256 : 1.71101),
 (-1.04532 : -0.403668)),
 ((0.648152 : 1.59426), (1.04326 : 1.33306), (0.921964 : 1.21235), (0.692053 : 1.40442),
 (-1.5269 : -0.980224)),
 ((0.516602 : 1.46866), (0.78724 : 1.5524), (0.900494 : 1.34625), (0.925993 : 1.67153),
 (-0.613392 : -0.408765)),
 ((0.674566 : 1.58438), (0.632099 : 1.55049), (0.760842 : 1.25024), (0.505051 : 1.39439),
 (-0.651067 : -0.468139)),
 ((0.650456 : 0.853465), (0.759926 : 1.70106), (1.1044 : 1.14576), (1.12409 : 1.15861),
 (-1.21589 : -0.671407)),
 ((0.915647 : 0.97702), (1.43216 : 1.55991), (1.07573 : 1.12609), (0.342677 : 1.00292),
 (-1.32554 : -1.00958)),
 ((0.948317 : 1.35885), (0.816691 : 1.03832), (1.09529 : 1.26174), (0.743721 : 1.25137),
 (-0.914808 : -0.479736)),
 ((0.334773 : 0.674261), (0.811609 : 0.940184), (0.960677 : 1.69398), (1.06948 : 1.71281),
 (-1.10665 : -0.592166)),
 ((0.865383 : 1.60256), (1.22417 : 1.69001), (0.807169 : 0.919355), (0.994049 : 1.81527),
 (-1.09955 : -0.990875)),
 ((0.988754 : 1.75013), (0.33079 : 0.747459), (1.40225 : 1.47), (0.559221 : 0.770501),
 (-1.67864 : -1.08753)),
 ((0.309214 : 0.927488), (0.734947 : 1.42525), (0.773736 : 1.68938), (1.02048 : 1.31898),
 (-0.88023 : -0.502045)),
 ((1.11124 : 1.88376), (0.619129 : 0.789697), (0.797647 : 0.978866), (0.983169 : 1.17879),
 (-0.634983 : -0.444731)),
 ((0.729728 : 1.03372), (0.521744 : 0.719901), (0.160649 : 0.931822), (0.820521 : 1.70156),

```

( -1.72254 : -1.27392 ) ),
(( 0.525605 : 1.32714 ), ( 0.713919 : 1.20878 ), ( 1.13013 : 1.36863 ), ( 1.01303 : 1.10578 ),
( -1.02489 : -0.131062 ) ),
(( 1.3272 : 1.38684 ), ( 0.674871 : 1.30691 ), ( 0.869182 : 1.62525 ), ( 0.30079 : 1.2107 ),
( -0.807825 : -0.608203 ) ),
(( 1.02722 : 1.07846 ), ( 0.626988 : 1.41736 ), ( 1.19213 : 1.21618 ), ( 0.955061 : 1.50842 ),
( -1.02356 : -0.148015 ) ),
(( 1.14422 : 1.44964 ), ( 0.653996 : 1.13588 ), ( 0.2266 : 0.878567 ), ( 0.116565 : 0.885144 ),
( -0.853679 : -0.633946 ) ),
(( 0.625996 : 0.683523 ), ( 1.15427 : 1.18696 ), ( 0.496933 : 1.1948 ), ( 1.08002 : 1.51628 ),
( -0.836055 : -0.200507 ) ),
(( 0.55681 : 1.03378 ), ( 1.33471 : 1.414 ), ( 0.529374 : 0.672597 ), ( 0.184011 : 1.10201 ),
( -1.37819 : -1.00436 ) ),
(( 0.055559 : 0.947188 ), ( 0.966948 : 1.51003 ), ( 1.09996 : 1.69965 ), ( -1.45805 : -0.651418 ),
( 0.547212 : 1.10619 ) ),
(( 0.239448 : 1.09973 ), ( 0.876873 : 0.895367 ), ( 1.2092 : 1.76332 ), ( -1.05353 : -0.126774 ),
( 0.261269 : 1.14512 ) ),
(( 0.377819 : 1.29597 ), ( 0.967956 : 1.18668 ), ( 0.596896 : 1.31225 ), ( -1.35504 : -1.28106 ),
( 1.29211 : 1.52136 ) ),
(( 0.603778 : 0.911863 ), ( 1.30744 : 1.44444 ), ( 0.814493 : 1.38177 ), ( -0.763985 : -0.543886 ),
( 0.870907 : 1.44835 ) ),
(( 1.22897 : 1.47238 ), ( 0.397702 : 0.703253 ), ( 0.286157 : 0.942671 ), ( -1.20911 : -0.687887 ),
( 0.543947 : 1.44235 ) ),
(( 0.431516 : 1.22146 ), ( 0.825236 : 1.41241 ), ( 0.764031 : 1.1934 ), ( -1.5316 : -1.15601 ),
( 0.87933 : 1.33891 ) ),
(( 0.199881 : 0.953749 ), ( 0.993225 : 1.07199 ), ( 0.566286 : 1.14736 ), ( -1.03874 : -0.343043 ),
( 0.191794 : 0.818155 ) ),
(( 0.720115 : 0.769829 ), ( 0.876461 : 1.77746 ), ( 0.296564 : 0.746956 ), ( -0.699316 : -0.636509 ),
( 0.774239 : 1.44137 ) ),
(( 0.852107 : 1.52641 ), ( 0.358974 : 0.8596 ), ( 0.339991 : 0.934919 ), ( -1.56601 : -0.998856 ),
( 1.11286 : 1.40318 ) ),
(( 0.744011 : 1.07108 ), ( 0.345256 : 0.683126 ), ( 0.600757 : 1.4893 ), ( -0.990982 : -0.079455 ),
( 0.749092 : 1.36715 ) ),
(( 0.794626 : 1.32034 ), ( 0.785058 : 0.873959 ), ( 0.998627 : 1.8196 ), ( -1.23371 : -0.835231 ),
( 0.64214 : 1.31346 ) ),
(( 0.405438 : 0.791772 ), ( 0.836894 : 1.23217 ), ( 0.487228 : 1.27091 ), ( -0.836543 : -0.484756 ),
( 0.899823 : 1.72553 ) ),
(( 1.07823 : 1.78022 ), ( 0.420804 : 0.779092 ), ( 1.11406 : 1.73243 ), ( -1.35078 : -0.38081 ),
( 0.390576 : 1.11292 ) ),
(( 0.723121 : 0.903424 ), ( 0.699835 : 0.731513 ), ( 1.21429 : 1.43103 ), ( -0.81753 : -0.61858 ),
( 0.340144 : 0.750008 ) ),
(( 0.8605 : 1.18256 ), ( 0.883892 : 1.6236 ), ( 0.542314 : 0.650807 ), ( -1.53717 : -0.973983 ),
( 0.365062 : 1.03943 ) ),
(( 0.536592 : 0.591403 ), ( 0.4906 : 1.24984 ), ( 0.79461 : 1.18601 ), ( -1.27975 : -0.641301 ),
( 1.31451 : 1.52649 ) ),
(( 1.04419 : 1.55797 ), ( 1.13149 : 1.51614 ), ( 0.391003 : 1.05512 ), ( -1.39053 : -0.779305 ),
( 0.593173 : 0.729896 ) ),
(( 0.754662 : 1.65548 ), ( 1.33386 : 1.59522 ), ( 0.362011 : 0.806207 ), ( -1.40938 : -0.642048 ),
( 0.135517 : 1.05498 ) ),
(( 0.342433 : 1.12804 ), ( 0.672628 : 1.4333 ), ( 1.05425 : 1.57129 ), ( -1.04659 : -0.625797 ),
( 0.205969 : 1.05362 ) ),
(( 0.725639 : 0.885403 ), ( 0.381024 : 1.0734 ), ( 0.425459 : 0.814966 ), ( -1.32028 : -0.971328 ),
( 1.02544 : 1.13195 ) ),
(( 0.974059 : 1.61644 ), ( 0.949324 : 1.39312 ), ( 0.768288 : 0.910138 ), ( -1.55461 : -0.681448 ),
( 1.1821 : 1.41347 ) ),
(( 1.05817 : 1.14725 ), ( 0.821253 : 1.13587 ), ( 1.08927 : 1.19483 ), ( -1.94855 : -1.01196 ),
( 1.18839 : 1.78549 ) ),
(( 0.941481 : 1.04311 ), ( 0.964736 : 0.979507 ), ( 0.3905 : 0.961318 ), ( -1.24052 : -0.730888 ),
( 1.20585 : 1.62435 ) ),
(( 0.970077 : 1.16326 ), ( 0.646489 : 1.27664 ), ( 0.566851 : 1.33448 ), ( -0.801798 : -0.779458 ),
( 1.1928 : 1.24889 ) ),
(( 0.595202 : 0.86166 ), ( 0.986724 : 1.45161 ), ( 0.355327 : 0.652181 ), ( -1.47903 : -0.570864 ),
( 0.842753 : 1.72803 ) ),
(( 0.493728 : 0.720878 ), ( 1.12513 : 1.78753 ), ( 1.06711 : 1.70174 ), ( -1.3599 : -0.99176 ),
( 0.991104 : 1.20699 ) ),
(( 0.937162 : 1.54848 ), ( 0.904523 : 0.922407 ), ( 1.01004 : 1.93069 ), ( -1.04685 : -0.28959 ),
( 0.985931 : 1.35389 ) ),

```


((1.11602 : 1.79487), (1.18651 : 1.73133), (0.880306 : 1.47624), (-1.17335 : -0.99942),
 (1.15542 : 1.79869)),
 ((1.36671 : 1.50346), (1.0669 : 1.84106), (0.78338 : 0.981536), (-0.97882 : -0.44557),
 (0.775506 : 0.985076)),
 ((0.68096 : 0.816492), (0.669729 : 0.741844), (1.2429 : 1.62854), (-1.4577 : -1.21703),
 (1.15708 : 1.61104)),
 ((0.822092 : 0.872082), (1.18004 : 1.26342), (0.390667 : 1.03537), (-0.598804 : -0.411267),
 (0.945311 : 1.73907)),
 ((1.13332 : 1.57529), (0.640049 : 1.01109), (0.808374 : 1.66585), (-1.72915 : -0.945769),
 (0.268395 : 1.14922)),
 ((0.520386 : 0.646977), (1.09291 : 1.14492), (0.527726 : 0.879696), (-1.14747 : -0.397778),
 (1.29916 : 1.52243)),
 ((1.43074 : 1.48601), (0.725333 : 1.07138), (0.802942 : 1.07099), (-1.27947 : -1.22869),
 (0.91464 : 1.01593)),
 ((1.24949 : 1.65993), (1.04855 : 1.1402), (1.25761 : 1.4319), (-0.678213 : -0.425489),
 (0.270333 : 1.16907)),
 ((1.20354 : 1.70728), (0.676733 : 0.73368), (0.193777 : 1.09877), (-1.53659 : -1.12844),
 (0.55385 : 0.785272)),
 ((0.414457 : 0.822184), (0.729331 : 1.64431), (0.042467 : 0.982437), (-1.64408 : -0.756661),
 (0.767037 : 0.920972)),
 ((0.489196 : 0.844188), (1.10552 : 1.78999), (0.948119 : 0.984039), (-1.10759 : -1.06871),
 (0.413892 : 1.30573)),
 ((0.292627 : 0.723853), (0.674291 : 1.55896), (0.624454 : 1.46915), (-1.34632 : -0.555864),
 (0.257149 : 0.861324)),
 ((0.620441 : 1.22114), (1.10025 : 1.33992), (1.21122 : 1.75423), (-1.05153 : -0.129688),
 (0.663427 : 0.915571)),
 ((1.19929 : 1.46055), (0.896496 : 1.01924), (0.707846 : 1.39027), (-1.16787 : -0.793222),
 (1.29551 : 1.65908)),
 ((0.094607 : 0.957518), (0.676778 : 1.35084), (0.981567 : 1.14408), (-0.678274 : -0.438307),
 (0.84579 : 1.80636)),
 ((0.517518 : 1.22732), (0.778985 : 1.06436), (1.14672 : 1.21752), (-1.76225 : -1.15657),
 (0.446287 : 1.23428)),
 ((0.502777 : 0.832194), (0.264367 : 0.810526), (1.11121 : 1.84607), (-1.92093 : -1.04367),
 (0.653844 : 1.51776)),
 ((0.796106 : 1.24934), (0.576937 : 0.987228), (0.589877 : 0.916547), (-0.620579 : -0.488067),
 (0.304758 : 1.20411)),
 ((0.346324 : 1.17011), (0.181082 : 1.08217), (1.20925 : 1.57901), (-1.57793 : -1.41865),
 (0.373638 : 1.29911)),
 ((0.367168 : 0.808191), (0.604633 : 1.34712), (0.787729 : 0.935926), (-1.1888 : -0.510926),
 (0.569979 : 0.717994)),
 ((0.319651 : 1.23573), (0.301584 : 0.820673), (0.777032 : 1.71172), (-1.47285 : -0.978057),
 (1.07498 : 1.32099)),
 ((0.688009 : 1.07355), (0.434797 : 1.37962), (1.01271 : 1.77299), (-1.45581 : -0.861492),
 (0.872784 : 1.54218)),
 ((0.886517 : 1.36767), (0.870418 : 1.06015), (0.652745 : 1.42358), (-1.32728 : -1.25401),
 (0.877514 : 1.47214)),
 ((0.359279 : 1.13952), (1.18905 : 1.78874), (0.797433 : 1.47598), (-1.68589 : -0.692938),
 (0.31193 : 1.00235)),
 ((0.728553 : 1.33923), (0.663915 : 0.831462), (0.593677 : 0.638264), (-0.894787 : -0.6189),
 (0.369884 : 0.725639)),
 ((1.02344 : 1.69506), (1.2472 : 1.85496), (0.443419 : 1.19271), (-1.21612 : -0.865825),
 (0.400937 : 0.644475)),
 ((0.894192 : 1.17841), (0.453993 : 0.919492), (1.07463 : 1.7143), (-1.65274 : -0.697455),
 (0.510605 : 0.808802)),
 ((0.095767 : 0.927488), (0.95352 : 1.944), (1.00316 : 1.62378), (-0.991287 : -0.990982),
 (0.379635 : 1.22237)),
 ((0.225517 : 0.932569), (0.976608 : 1.11943), (0.629612 : 1.62616), (-1.59827 : -1.35729),
 (0.588031 : 0.660939)),
 ((0.532105 : 1.01943), (1.14847 : 1.85043), (0.689199 : 0.990875), (-0.757866 : -0.34077),
 (0.592715 : 0.87196)),
 ((0.89201 : 1.61188), (0.858287 : 1.10146), (0.316401 : 1.12191), (-1.44558 : -1.41934),
 (1.06862 : 1.91411)),
 ((1.08518 : 1.34699), (0.603381 : 0.960845), (0.913099 : 1.87919), (-1.58913 : -0.894131),
 (0.889096 : 1.08145)),
 ((0.608325 : 0.795312), (0.704001 : 1.24323), (0.130696 : 1.09014), (-1.27191 : -0.478912),
 (0.910459 : 1.87982)),
 ((0.753731 : 1.24032), (1.18697 : 1.47049), (0.907163 : 1.56267), (-1.16312 : -1.03089),

(0.244636 : 0.816828),
 ((0.365337 : 1.13504), (0.79902 : 1.53946), (0.448912 : 1.12136), (-1.90819 : -0.977493),
 (1.17121 : 1.27805),
 ((0.377239 : 0.914579), (0.340938 : 0.938002), (0.632572 : 0.672735), (-1.48558 : -1.14116),
 (0.561983 : 1.4118),
 ((0.251564 : 1.24155), (0.628224 : 0.948027), (0.827219 : 0.998306), (-1.12169 : -0.916059),
 (0.487701 : 1.19103),
 ((1.07883 : 1.43123), (0.693915 : 0.812876), (0.426603 : 1.31872), (-1.33929 : -0.665563),
 (0.349879 : 1.08202),
 ((0.193854 : 1.18604), (1.16613 : 1.78211), (0.235206 : 1.09909), (-0.888089 : -0.350871),
 (1.25622 : 1.47723),
 ((1.01041 : 1.79916), (0.390637 : 1.21955), (1.09236 : 1.51987), (-1.05232 : -0.331599),
 (1.23569 : 1.67251),
 ((0.694815 : 1.60247), (0.64008 : 1.00722), (0.552553 : 0.948012), (-1.29855 : -0.561953),
 (0.590381 : 0.629658),
 ((0.602359 : 1.57468), (0.465346 : 0.732536), (0.88377 : 1.05834), (-1.30635 : -1.10184),
 (0.514145 : 0.997955),
 ((0.395764 : 0.625294), (0.720878 : 1.05362), (0.405896 : 0.898068), (-1.39976 : -1.0351),
 (0.886334 : 1.31311),
 ((1.27143 : 1.56993), (0.83723 : 1.62766), (1.08969 : 1.61263), (-0.857479 : -0.599811),
 (0.685217 : 1.38513),
 ((0.542787 : 0.646123), (0.84257 : 1.00182), (0.675222 : 0.679647), (-1.01358 : -0.647816),
 (0.57181 : 1.06407),
 ((1.28968 : 1.52916), (1.25599 : 1.34257), (1.16913 : 1.78008), (-1.17128 : -0.529115),
 (1.18436 : 1.77682),
 ((0.684362 : 1.62427), (1.10578 : 1.25217), (1.24706 : 1.3001), (-1.33302 : -0.700522),
 (0.868221 : 1.38847),
 ((0.267632 : 0.88197), (0.275826 : 1.13227), (1.14191 : 1.84918), (-1.28378 : -0.851146),
 (0.258263 : 1.00664),
 ((1.17843 : 1.43509), (0.966887 : 1.03705), (0.842814 : 1.52777), (-1.34103 : -1.18227),
 (0.643559 : 1.24105),
 ((0.455077 : 0.933363), (0.585986 : 0.652913), (0.187048 : 0.929533), (-0.790384 : -0.73722),
 (0.689505 : 1.10434),
 ((1.33721 : 1.49139), (0.666646 : 1.38508), (1.00066 : 1.94588), (-1.35008 : -0.41319),
 (0.184072 : 1.12093),
 ((1.31188 : 1.59937), (0.275277 : 0.9588), (1.15236 : 1.30032), (-1.13462 : -1.06092),
 (0.322367 : 1.24021),
 ((1.28811 : 1.62107), (0.601062 : 0.896939), (0.538591 : 1.3097), (-1.20821 : -0.953261),
 (0.463301 : 1.26804),
 ((1.05287 : 1.54807), (0.360164 : 0.646794), (0.928266 : 1.58521), (-1.11867 : -0.445982),
 (1.09429 : 1.16918),
 ((1.37785 : 1.43611), (1.0877 : 1.44214), (0.57741 : 1.19227), (-1.46921 : -0.917982),
 (1.23592 : 1.69413),
 ((1.11946 : 1.35549), (0.65302 : 1.13277), (0.889004 : 1.11414), (-1.30725 : -1.13555),
 (1.0246 : 1.28288),
 ((0.199622 : 1.17753), (1.23086 : 1.6359), (0.950896 : 1.15192), (-1.57341 : -1.09253),
 (0.535234 : 1.53334),
 ((0.780755 : 1.03519), (0.799539 : 1.294), (0.854411 : 1.58124), (-1.24711 : -0.698172),
 (1.36505 : 1.62964),
 ((1.06977 : 1.26975), (0.838374 : 1.70592), (1.08226 : 1.40741), (-1.33627 : -0.984024),
 (0.902036 : 1.44646),
 ((0.318201 : 1.05138), (0.563295 : 1.46237), (0.275277 : 0.857418), (-1.17374 : -0.836604),
 (0.863918 : 1.51149),
 ((0.532807 : 1.06969), (0.593249 : 1.108), (0.654347 : 0.753502), (-1.29113 : -0.702673),
 (0.42291 : 1.06325),
 ((0.995972 : 1.1749), (0.18543 : 1.00232), (0.095462 : 1.06824), (-1.12723 : -0.314432),
 (0.348277 : 1.01795),
 ((1.22327 : 1.54436), (1.4335 : 1.50009), (0.652058 : 1.20295), (-1.18377 : -0.813272),
 (0.406339 : 0.69718),
 ((1.2707 : 1.29118), (0.897549 : 1.45839), (0.993851 : 1.79527), (-1.64934 : -0.766961),
 (0.575976 : 1.10544),
 ((0.161824 : 0.861675), (1.34271 : 1.43408), (0.10448 : 1.08748), (-0.664754 : -0.546434),
 (0.274667 : 0.817194),
 ((0.943724 : 1.01974), (0.916272 : 1.00618), (0.959426 : 1.01119), (-0.983215 : -0.410901),
 (1.06182 : 1.08214),
 ((0.814341 : 1.62876), (0.505325 : 1.37477), (0.106189 : 1.10408), (-1.47734 : -0.511093),
 (0.999893 : 1.01027),

((1.20879 : 1.75495), (0.497116 : 0.69161), (0.902402 : 1.60765), (-0.948256 : -0.419675),
 (1.33023 : 1.60739)),
 ((0.991256 : 1.76173), (1.02925 : 1.40555), (0.44406 : 1.03929), (-1.18685 : -0.2987),
 (0.661519 : 0.920042)),
 ((0.798242 : 1.4864), (0.376476 : 1.19269), (0.254585 : 1.125), (-0.91142 : -0.479156),
 (0.798013 : 1.56525)),
 ((0.916578 : 1.72227), (1.18845 : 1.22956), (0.909406 : 1.15697), (-1.74961 : -1.1194),
 (0.954939 : 1.3952)),
 ((0.681143 : 1.50957), (0.678884 : 1.0878), (0.76223 : 1.04389), (-1.52716 : -0.789468),
 (0.73986 : 0.921293)),
 ((0.543458 : 1.10433), (0.389889 : 0.826609), (1.12227 : 1.48324), (-1.46686 : -1.36114),
 (1.22758 : 1.33204)),
 ((0.391659 : 0.61127), (0.644917 : 0.922361), (0.354808 : 1.07434), (-1.3184 : -0.601123),
 (-0.855052 : -0.483703)),
 ((0.23426 : 1.09589), (0.300012 : 0.823466), (1.12192 : 1.19514), (-1.01627 : -0.527207),
 (-1.54181 : -0.940046)),
 ((0.989975 : 1.38055), (0.319117 : 0.75193), (1.37382 : 1.42665), (-0.956114 : -0.379345),
 (-1.71401 : -0.862636)),
 ((0.372829 : 1.32598), (0.340114 : 1.26202), (0.375774 : 0.902615), (-0.910001 : -0.272378),
 (-1.12709 : -0.274163)),
 ((0.945296 : 1.70545), (0.302377 : 0.987793), (1.10421 : 1.27486), (-1.26267 : -0.45555),
 (-1.66953 : -1.08235)),
 ((1.00519 : 1.74145), (0.653508 : 0.828196), (0.612781 : 1.07907), (-1.68078 : -1.22687),
 (-1.56963 : -0.674184)),
 ((1.07027 : 1.48532), (0.837336 : 0.938688), (1.08318 : 1.32653), (-1.21697 : -1.0985),
 (-1.35873 : -0.797189)),
 ((0.965896 : 1.33883), (0.370647 : 0.967254), (0.583651 : 1.01598), (-1.38205 : -1.20553),
 (-1.14183 : -0.915876)),
 ((0.7978 : 1.76115), (0.7875 : 1.17975), (0.788263 : 1.61681), (-1.67106 : -0.761849),
 (-1.79801 : -1.02022)),
 ((0.465896 : 1.13007), (0.611728 : 0.941939), (1.3612 : 1.48096), (-1.44212 : -0.682669),
 (-1.44914 : -0.717828)),
 ((1.41652 : 1.51836), (0.8811 : 1.15378), (0.45584 : 0.794717), (-1.35879 : -0.632206),
 (-0.840724 : -0.181249)),
 ((0.09624 : 0.991714), (0.643117 : 0.85165), (0.872082 : 1.2615), (-0.752602 : -0.524323),
 (-1.10398 : -0.956542)),
 ((0.127216 : 1.1256), (1.34643 : 1.49426), (0.423795 : 0.904645), (-1.1858 : -0.721213),
 (-1.15795 : -1.0811)),
 ((1.12917 : 1.45468), (1.21419 : 1.6146), (0.554704 : 0.88348), (-1.12371 : -0.64713),
 (-1.1962 : -0.722465)),
 ((0.936766 : 1.20591), (1.07926 : 1.24177), (0.764489 : 1.70962), (-1.40834 : -0.579791),
 (-1.22536 : -0.443907)),
 ((1.25787 : 1.6062), (1.34066 : 1.46145), (0.533418 : 1.42608), (-1.58968 : -1.30741),
 (-1.4183 : -1.04488)),
 ((0.154042 : 0.987564), (1.23331 : 1.37587), (1.22337 : 1.35371), (-0.561525 : -0.462462),
 (-1.55202 : -1.02164)),
 ((0.165731 : 1.02629), (1.04972 : 1.23615), (0.45027 : 0.605579), (-1.7994 : -0.972747),
 (-0.95555 : -0.863872)),
 ((0.504379 : 0.93112), (0.374126 : 0.834437), (1.22857 : 1.41003), (-1.74595 : -1.01997),
 (-1.1341 : -0.453078)),
 ((0.564714 : 1.42976), (0.852886 : 1.55152), (0.844462 : 1.54962), (-1.47627 : -0.980316),
 (-1.50021 : -0.577517)),
 ((0.627644 : 0.679922), (1.08707 : 1.44209), (0.487945 : 0.611484), (-1.63181 : -0.848521),
 (-0.869167 : -0.331217)),
 ((0.094333 : 0.936552), (0.961715 : 1.59583), (1.22008 : 1.43323), (-0.681967 : -0.434004),
 (-0.998795 : -0.242363)),
 ((0.475692 : 0.630512), (0.179861 : 0.919385), (0.438215 : 0.810816), (-1.29075 : -0.473495),
 (-1.41435 : -1.18146)),
 ((0.721747 : 1.64176), (0.512528 : 1.13813), (0.19335 : 0.899792), (-0.867977 : -0.544145),
 (-0.992599 : -0.255516)),
 ((0.375057 : 1.17013), (1.37089 : 1.39558), (1.06226 : 1.46703), (-1.21355 : -1.09575),
 (-0.723426 : -0.304224)),
 ((1.03322 : 1.40713), (0.3155 : 1.02826), (0.359401 : 0.937544), (-0.949721 : -0.362545),
 (-1.15995 : -0.595874)),
 ((0.770104 : 1.37806), (0.102954 : 1.09554), (0.340541 : 1.20531), (-1.62417 : -0.678854),
 (-1.74703 : -1.18503)),
 ((1.05797 : 1.54663), (0.393918 : 0.739204), (0.269158 : 1.13411), (-0.887738 : -0.616489),

(-1.42021 : -0.953673),
 ((0.412442 : 1.23772), (0.874187 : 1.12377), (0.351802 : 1.24691), (-1.4488 : -0.705344),
 (-1.53006 : -1.42483)),
 ((1.01482 : 1.32684), (0.828639 : 1.24787), (0.657735 : 1.48103), (-1.50974 : -1.09413),
 (-0.708533 : -0.325159)),
 ((0.512589 : 0.656484), (0.725623 : 1.01677), (0.788537 : 1.75582), (-1.04996 : -0.179907),
 (-1.19137 : -0.894177)),
 ((1.1222 : 1.455), (1.17455 : 1.80213), (1.31378 : 1.6496), (-1.0584 : -0.284997),
 (-0.781243 : -0.40199)),
 ((0.618748 : 1.33978), (0.777093 : 1.48595), (0.4035 : 1.39962), (-1.58408 : -0.70777),
 (-1.01915 : -0.064104)),
 ((0.36256 : 0.976257), (0.463225 : 1.36999), (0.73162 : 0.910611), (-1.19921 : -0.728492),
 (-1.16019 : -0.331828)),
 ((1.23895 : 1.31658), (0.451353 : 1.02272), (1.15232 : 1.17103), (-1.09276 : -0.226463),
 (-1.66742 : -0.717017)),
 ((0.341182 : 0.751778), (1.05879 : 1.31875), (0.43611 : 1.11963), (-0.972488 : -0.773324),
 (-1.57686 : -1.17997)),
 ((0.47203 : 0.663411), (0.277367 : 0.787881), (1.04791 : 1.26682), (-1.18706 : -0.219138),
 (-1.68693 : -0.990127)),
 ((0.630528 : 1.43994), (0.881268 : 1.62006), (0.62064 : 1.43884), (-0.924177 : -0.701575),
 (-1.02446 : -0.668615)),
 ((0.663076 : 1.22324), (0.913282 : 1.45706), (0.992889 : 1.57161), (-1.43329 : -0.987106),
 (-0.875866 : -0.455626)),
 ((1.2405 : 1.65226), (0.764321 : 1.62842), (0.989593 : 1.21894), (-1.39979 : -0.419691),
 (-1.27784 : -1.0231)),
 ((0.348521 : 0.866268), (1.27987 : 1.44394), (0.604434 : 1.54251), (-0.800989 : -0.310465),
 (-1.68419 : -1.18982)),
 ((1.10524 : 1.70591), (0.462004 : 0.729957), (0.445387 : 0.711722), (-1.23565 : -0.670385),
 (-1.49069 : -1.00234)),
 ((1.00745 : 1.58989), (0.444746 : 1.17344), (0.798685 : 1.54267), (-0.781457 : -0.57445),
 (-1.65561 : -1.03444)),
 ((1.22741 : 1.64515), (0.570818 : 1.40797), (0.511032 : 1.03543), (-1.33216 : -0.46791),
 (-1.729 : -0.942137)),
 ((0.883724 : 1.00406), (1.14847 : 1.82662), (0.6489 : 0.919233), (-1.03647 : -0.189367),
 (-1.14275 : -1.00401)),
 ((0.870052 : 1.45274), (0.882183 : 1.02226), (0.128346 : 1.0652), (-1.34321 : -0.710227),
 (-1.05074 : -0.889294)),
 ((0.895322 : 0.999603), (1.11254 : 1.72883), (0.284219 : 0.87405), (-1.68706 : -0.883297),
 (-1.27854 : -1.04126)),
 ((0.436949 : 1.12386), (0.431379 : 0.849574), (0.223029 : 1.16993), (-1.46602 : -1.17093),
 (-1.49072 : -1.15465)),
 ((0.896191 : 1.14583), (1.02057 : 1.80377), (1.34365 : 1.42245), (-1.10067 : -0.694098),
 (-1.43901 : -1.07465)),
 ((0.93234 : 1.02429), (0.758477 : 1.68139), (0.176977 : 0.968047), (-1.56624 : -0.824366),
 (-1.55985 : -1.36319)),
 ((0.806024 : 1.11441), (0.812632 : 1.76432), (0.251518 : 0.988296), (-0.787576 : -0.740761),
 (-1.68198 : -1.08077)),
 ((0.639927 : 1.20452), (0.646443 : 1.52608), (0.598209 : 1.02321), (-1.62217 : -0.706732),
 (-1.11867 : -0.829234)),
 ((0.874142 : 1.67681), (0.683981 : 0.963073), (0.830012 : 0.931608), (-1.26418 : -0.600223),
 (-1.39505 : -1.10451)),
 ((0.412748 : 0.591769), (0.852748 : 1.71279), (1.02951 : 1.1529), (-1.72268 : -1.26646),
 (-1.2344 : -0.335322)),
 ((0.592486 : 1.57149), (0.928358 : 1.07357), (0.513855 : 0.555483), (-1.49405 : -1.36941),
 (-1.63869 : -0.940641)),
 ((1.21361 : 1.732), (0.59978 : 1.4434), (0.986129 : 1.27068), (-0.603336 : -0.40698),
 (-0.880276 : -0.142064)),
 ((1.02402 : 1.28846), (0.516511 : 1.0244), (0.683523 : 0.889492), (-0.765297 : -0.239708),
 (-1.91739 : -0.929472)),
 ((0.29725 : 1.0033), (0.339396 : 1.08939), (0.963057 : 1.43475), (-0.955016 : -0.175115),
 (-1.58765 : -0.80784)),
 ((0.561296 : 0.567309), (0.551103 : 1.35859), (0.882092 : 1.29629), (-1.11382 : -0.162374),
 (-1.36425 : -1.06993)),
 ((0.674978 : 0.973174), (1.05774 : 1.78112), (1.30288 : 1.52445), (-1.48909 : -1.29795),
 (-0.909085 : -0.646168)),
 ((0.381909 : 1.15638), (0.987762 : 1.93912), (0.93144 : 1.69843), (-1.15186 : -0.432752),
 (-1.57305 : -1.40327)),

((0.25161 : 0.821024), (0.63564 : 1.58559), (1.06436 : 1.71645), (-0.99411 : -0.521409),
 (-1.30273 : -0.898785)),
 ((0.615329 : 0.629185), (1.2636 : 1.67322), (0.213599 : 0.972015), (-0.911206 : -0.271538),
 (-1.06192 : -0.870174)),
 ((1.27651 : 1.49795), (0.53029 : 0.820063), (1.28295 : 1.68287), (-1.36805 : -0.95407),
 (-1.29662 : -0.452849)),
 ((0.987487 : 1.30552), (0.223563 : 1.17727), (1.13277 : 1.30441), (-1.06047 : -0.731666),
 (-1.10346 : -1.09488)),
 ((0.998199 : 1.04578), (0.793985 : 1.29864), (0.76307 : 1.68247), (-0.860317 : -0.36845),
 (-1.54309 : -1.18011)),
 ((0.458708 : 0.574007), (0.485397 : 0.553728), (0.523499 : 0.562441), (-0.605609 : -0.507736),
 (-0.905179 : -0.298868)),
 ((1.16414 : 1.74735), (1.19503 : 1.55759), (0.79371 : 1.4678), (-1.50819 : -0.663961),
 (-1.62165 : -0.726112)),
 ((1.12015 : 1.16279), (0.633229 : 1.20243), (1.02097 : 1.46858), (-0.933378 : -0.477706),
 (-0.875347 : -0.798074)),
 ((0.755531 : 1.02916), (0.45732 : 0.857814), (0.427244 : 0.679388), (-1.00211 : -0.443709),
 (-1.20074 : -0.828104)),
 ((0.864605 : 1.6551), (0.88142 : 1.33761), (0.658528 : 1.13111), (-1.80696 : -1.05364),
 (-0.957549 : -0.285501)),
 ((0.598285 : 0.789605), (0.383297 : 1.01015), (0.212378 : 1.14914), (-1.46388 : -1.40898),
 (-0.916959 : -0.489059)),
 ((0.597842 : 1.29733), (0.598544 : 1.13347), (1.20887 : 1.22617), (-1.17275 : -1.17019),
 (-0.769585 : -0.353893)),
 ((0.240379 : 1.07317), (0.376507 : 1.32078), (1.02187 : 1.78471), (-1.37236 : -0.52385),
 (-1.58164 : -1.08632)),
 ((0.50502 : 1.3957), (0.832316 : 1.66895), (0.533311 : 1.37779), (-0.888699 : -0.489181),
 (-1.53001 : -1.4601)),
 ((1.23679 : 1.26356), (1.12923 : 1.70377), (0.865459 : 1.45691), (-1.12102 : -0.654027),
 (-1.07372 : -0.743141)),
 ((1.13868 : 1.34843), (0.726157 : 1.09442), (0.456328 : 1.03301), (-1.23231 : -0.647908),
 (-1.50737 : -0.785333)),
 ((0.73191 : 1.21721), (0.990524 : 1.2575), (1.32917 : 1.56212), (-1.44186 : -0.63184),
 (-1.52883 : -1.0408)),
 ((0.343623 : 1.11098), (0.46791 : 0.692068), (0.429319 : 1.37912), (-0.798563 : -0.629551),
 (-0.976012 : -0.194037)),
 ((1.48077 : 1.50922), (0.280908 : 0.851604), (1.03062 : 1.18084), (-1.67948 : -0.95143),
 (-1.1832 : -0.399762)),
 ((0.784402 : 1.14049), (0.661672 : 1.25144), (0.500504 : 0.599536), (-1.64696 : -1.08374),
 (-0.913678 : -0.244469)),
 ((0.698904 : 0.712912), (0.568865 : 1.52193), (0.732124 : 1.5054), (-0.650624 : -0.546892),
 (-1.48286 : -1.20811)),
 ((1.09073 : 1.61672), (0.262764 : 0.932432), (0.856014 : 1.0159), (-1.80654 : -1.17551),
 (-1.92392 : -0.954009)),
 ((1.3827 : 1.3864), (0.674429 : 0.719321), (0.501267 : 1.45762), (-1.07282 : -0.856594),
 (-1.81613 : -1.02457)),
 ((0.386929 : 0.986007), (0.966979 : 1.03751), (0.947996 : 1.91223), (-0.999268 : -0.790735),
 (-1.61887 : -1.27261)),
 ((0.48909 : 0.686026), (0.916776 : 1.71078), (1.06012 : 1.66906), (-1.79659 : -1.12662),
 (-0.98001 : -0.435652)),
 ((1.22079 : 1.62577), (0.194464 : 0.864803), (0.974868 : 1.08876), (-0.690115 : -0.676107),
 (-1.94664 : -0.967483)),
 ((1.11728 : 1.78054), (0.369854 : 1.17771), (1.21348 : 1.24531), (-1.32598 : -0.656713),
 (-1.23934 : -0.393735)),
 ((0.65714 : 0.675512), (1.0593 : 1.56447), (0.730293 : 0.942518), (-1.32397 : -0.3417),
 (-1.19675 : -0.371929)),
 ((0.081103 : 0.98616), (1.14284 : 1.81022), (0.496918 : 1.48651), (-1.86158 : -0.941816),
 (-1.08773 : -0.614689)),
 ((1.05261 : 1.33219), (0.464278 : 1.16929), (1.0855 : 1.62055), (-1.35789 : -0.804437),
 (-1.16941 : -0.70159)),
 ((1.01447 : 1.79589), (0.450407 : 0.942854), (0.935286 : 0.98558), (-1.85002 : -0.911817),
 (-1.57041 : -0.863536)),
 ((1.27619 : 1.65575), (1.21338 : 1.22263), (0.62122 : 1.54355), (-1.9496 : -1.00059),
 (-1.74143 : -0.891003)),
 ((0.398953 : 0.723731), (0.988922 : 1.52962), (0.551744 : 0.77633), (-1.27514 : -1.22375),
 (-1.77775 : -1.20559)),
 ((0.891903 : 1.18271), (0.780374 : 1.6745), (1.17467 : 1.34668), (-1.90959 : -0.945204),

```

( -1.81468 : -1.15413 ),
(( 1.15534 : 1.63659 ), ( 1.38363 : 1.50418 ), ( 1.01897 : 1.45337 ), ( -1.67455 : -0.808161 ),
( -0.86906 : -0.551668 )),
(( 1.06928 : 1.39433 ), ( 0.126011 : 0.968047 ), ( 0.817042 : 1.53383 ), ( -0.777917 : -0.46672 ),
( -1.3804 : -0.721442 )),
(( 1.18899 : 1.4949 ), ( 0.222877 : 0.823267 ), ( 0.092135 : 0.935331 ), ( -1.61583 : -1.12397 ),
( -1.2364 : -0.57564 )),
(( 0.23629 : 0.835551 ), ( 0.203467 : 0.988617 ), ( 0.527512 : 1.39299 ), ( -0.634312 : -0.595187 ),
( -1.18421 : -0.456862 )),
(( 0.530091 : 1.17489 ), ( 0.445097 : 1.23925 ), ( 0.715125 : 1.29382 ), ( -1.61771 : -1.32424 ),
( -1.66675 : -0.831584 ))
))
END

```

Fichier "chavent.txt"

400

5

"Ind_ 1"	2	3	3	3
"Ind_ 2"	2	3	3	3
"Ind_ 3"	2	3	3	3
"Ind_ 4"	2	3	3	3
"Ind_ 5"	2	3	3	3
"Ind_ 6"	2	3	3	3
"Ind_ 7"	2	3	3	3
"Ind_ 8"	2	3	3	3
"Ind_ 9"	2	3	3	3
"Ind_ 10"	2	3	3	3
"Ind_ 11"	2	3	3	3
"Ind_ 12"	2	3	3	3
"Ind_ 13"	2	3	3	3
"Ind_ 14"	2	3	3	3
"Ind_ 15"	2	3	3	3
"Ind_ 16"	2	3	3	3
"Ind_ 17"	2	3	3	3
"Ind_ 18"	2	3	3	3
"Ind_ 19"	2	3	3	3
"Ind_ 20"	2	3	3	3
"Ind_ 21"	2	3	3	3
"Ind_ 22"	2	3	3	3
"Ind_ 23"	2	3	3	3
"Ind_ 24"	2	3	3	3
"Ind_ 25"	2	3	3	3
"Ind_ 26"	2	3	3	3
"Ind_ 27"	2	3	3	3
"Ind_ 28"	2	3	3	3
"Ind_ 29"	2	3	3	3
"Ind_ 30"	2	3	3	3
"Ind_ 31"	2	3	3	3
"Ind_ 32"	2	3	3	3
"Ind_ 33"	2	3	3	3
"Ind_ 34"	2	3	3	3
"Ind_ 35"	2	3	3	3
"Ind_ 36"	2	3	3	3
"Ind_ 37"	2	3	3	3
"Ind_ 38"	2	3	3	3
"Ind_ 39"	2	3	3	3
"Ind_ 40"	2	3	3	3
"Ind_ 41"	2	3	3	3
"Ind_ 42"	2	3	3	3
"Ind_ 43"	2	3	3	3
"Ind_ 44"	2	3	3	3
"Ind_ 45"	2	3	3	3
"Ind_ 46"	2	3	3	3
"Ind_ 47"	2	3	3	3
"Ind_ 48"	2	3	3	3
"Ind_ 49"	2	3	3	3

"Ind_ 50"	2	3	3	3
"Ind_ 51"	2	3	3	3
"Ind_ 52"	2	3	3	3
"Ind_ 53"	2	3	3	3
"Ind_ 54"	2	3	3	3
"Ind_ 55"	2	3	3	3
"Ind_ 56"	2	3	3	3
"Ind_ 57"	2	3	3	3
"Ind_ 58"	2	3	3	3
"Ind_ 59"	2	3	3	3
"Ind_ 60"	2	3	3	3
"Ind_ 61"	2	3	3	3
"Ind_ 62"	2	3	3	3
"Ind_ 63"	2	3	3	3
"Ind_ 64"	2	3	3	3
"Ind_ 65"	2	3	3	3
"Ind_ 66"	2	3	3	3
"Ind_ 67"	2	3	3	3
"Ind_ 68"	2	3	3	3
"Ind_ 69"	2	3	3	3
"Ind_ 70"	2	3	3	3
"Ind_ 71"	2	3	3	3
"Ind_ 72"	2	3	3	3
"Ind_ 73"	2	3	3	3
"Ind_ 74"	2	3	3	3
"Ind_ 75"	2	3	3	3
"Ind_ 76"	2	3	3	3
"Ind_ 77"	2	3	3	3
"Ind_ 78"	2	3	3	3
"Ind_ 79"	2	3	3	3
"Ind_ 80"	2	3	3	3
"Ind_ 81"	2	3	3	3
"Ind_ 82"	2	3	3	3
"Ind_ 83"	2	3	3	3
"Ind_ 84"	2	3	3	3
"Ind_ 85"	2	3	3	3
"Ind_ 86"	2	3	3	3
"Ind_ 87"	2	3	3	3
"Ind_ 88"	2	3	3	3
"Ind_ 89"	2	3	3	3
"Ind_ 90"	2	3	3	3
"Ind_ 91"	2	3	3	3
"Ind_ 92"	2	3	3	3
"Ind_ 93"	2	3	3	3
"Ind_ 94"	2	3	3	3
"Ind_ 95"	2	3	3	3
"Ind_ 96"	2	3	3	3
"Ind_ 97"	2	3	3	3
"Ind_ 98"	2	3	3	3
"Ind_ 99"	2	3	3	3
"Ind_ 100"	2	3	3	3
"Ind_ 101"	1	1	4	4
"Ind_ 102"	1	1	4	4
"Ind_ 103"	1	1	4	4
"Ind_ 104"	1	1	4	4
"Ind_ 105"	1	1	4	4
"Ind_ 106"	1	1	4	4
"Ind_ 107"	1	1	4	4
"Ind_ 108"	1	1	4	4
"Ind_ 109"	1	1	4	4
"Ind_ 110"	1	1	4	4
"Ind_ 111"	1	1	4	4
"Ind_ 112"	1	1	4	4
"Ind_ 113"	1	1	4	4
"Ind_ 114"	1	1	4	4
"Ind_ 115"	1	1	4	4
"Ind_ 116"	1	1	4	4

"Ind_ 117"	1	1	4	4
"Ind_ 118"	1	1	4	4
"Ind_ 119"	1	1	4	4
"Ind_ 120"	1	1	4	4
"Ind_ 121"	1	1	4	4
"Ind_ 122"	1	1	4	4
"Ind_ 123"	1	1	4	4
"Ind_ 124"	1	1	4	4
"Ind_ 125"	1	1	4	4
"Ind_ 126"	1	1	4	4
"Ind_ 127"	1	1	4	4
"Ind_ 128"	1	1	4	4
"Ind_ 129"	1	1	4	4
"Ind_ 130"	1	1	4	4
"Ind_ 131"	1	1	4	4
"Ind_ 132"	1	1	4	4
"Ind_ 133"	1	1	4	4
"Ind_ 134"	1	1	4	4
"Ind_ 135"	1	1	4	4
"Ind_ 136"	1	1	4	4
"Ind_ 137"	1	1	4	4
"Ind_ 138"	1	1	4	4
"Ind_ 139"	1	1	4	4
"Ind_ 140"	1	1	4	4
"Ind_ 141"	1	1	4	4
"Ind_ 142"	1	1	4	4
"Ind_ 143"	1	1	4	4
"Ind_ 144"	1	1	4	4
"Ind_ 145"	1	1	4	4
"Ind_ 146"	1	1	4	4
"Ind_ 147"	1	1	4	4
"Ind_ 148"	1	1	4	4
"Ind_ 149"	1	1	4	4
"Ind_ 150"	1	1	4	4
"Ind_ 151"	1	1	4	4
"Ind_ 152"	1	1	4	4
"Ind_ 153"	1	1	4	4
"Ind_ 154"	1	1	4	4
"Ind_ 155"	1	1	4	4
"Ind_ 156"	1	1	4	4
"Ind_ 157"	1	1	4	4
"Ind_ 158"	1	1	4	4
"Ind_ 159"	1	1	4	4
"Ind_ 160"	1	1	4	4
"Ind_ 161"	1	1	4	4
"Ind_ 162"	1	1	4	4
"Ind_ 163"	1	1	4	4
"Ind_ 164"	1	1	4	4
"Ind_ 165"	1	1	4	4
"Ind_ 166"	1	1	4	4
"Ind_ 167"	1	1	4	4
"Ind_ 168"	1	1	4	4
"Ind_ 169"	1	1	4	4
"Ind_ 170"	1	1	4	4
"Ind_ 171"	1	1	4	4
"Ind_ 172"	1	1	4	4
"Ind_ 173"	1	1	4	4
"Ind_ 174"	1	1	4	4
"Ind_ 175"	1	1	4	4
"Ind_ 176"	1	1	4	4
"Ind_ 177"	1	1	4	4
"Ind_ 178"	1	1	4	4
"Ind_ 179"	1	1	4	4
"Ind_ 180"	1	1	4	4
"Ind_ 181"	1	1	4	4
"Ind_ 182"	1	1	4	4
"Ind_ 183"	1	1	4	4

"Ind_ 318"	1	1	1	1
"Ind_ 319"	1	1	1	1
"Ind_ 320"	1	1	1	1
"Ind_ 321"	1	1	1	1
"Ind_ 322"	1	1	1	1
"Ind_ 323"	1	1	1	1
"Ind_ 324"	1	1	1	5
"Ind_ 325"	1	1	1	1
"Ind_ 326"	1	1	1	5
"Ind_ 327"	1	1	1	5
"Ind_ 328"	1	1	1	5
"Ind_ 329"	1	1	1	1
"Ind_ 330"	1	1	1	5
"Ind_ 331"	1	1	1	1
"Ind_ 332"	1	1	1	5
"Ind_ 333"	1	1	1	1
"Ind_ 334"	1	1	1	1
"Ind_ 335"	1	1	1	5
"Ind_ 336"	1	1	1	1
"Ind_ 337"	1	1	1	1
"Ind_ 338"	1	1	1	1
"Ind_ 339"	1	1	1	1
"Ind_ 340"	1	1	1	5
"Ind_ 341"	1	1	1	1
"Ind_ 342"	1	1	1	5
"Ind_ 343"	1	1	1	5
"Ind_ 344"	1	1	1	5
"Ind_ 345"	1	1	1	1
"Ind_ 346"	1	1	1	5
"Ind_ 347"	1	1	1	1
"Ind_ 348"	1	1	1	1
"Ind_ 349"	1	1	1	1
"Ind_ 350"	1	1	1	1
"Ind_ 351"	1	1	1	1
"Ind_ 352"	1	1	1	1
"Ind_ 353"	1	1	1	5
"Ind_ 354"	1	1	1	1
"Ind_ 355"	1	1	1	5
"Ind_ 356"	1	1	1	5
"Ind_ 357"	1	1	1	5
"Ind_ 358"	1	1	1	1
"Ind_ 359"	1	1	1	1
"Ind_ 360"	1	1	1	1
"Ind_ 361"	1	1	1	1
"Ind_ 362"	1	1	1	1
"Ind_ 363"	1	1	1	1
"Ind_ 364"	1	1	1	5
"Ind_ 365"	1	1	1	5
"Ind_ 366"	1	1	1	1
"Ind_ 367"	1	1	1	1
"Ind_ 368"	1	1	1	5
"Ind_ 369"	1	1	1	5
"Ind_ 370"	1	1	1	1
"Ind_ 371"	1	1	1	5
"Ind_ 372"	1	1	1	1
"Ind_ 373"	1	1	1	1
"Ind_ 374"	1	1	1	1
"Ind_ 375"	1	1	1	1
"Ind_ 376"	1	1	1	5
"Ind_ 377"	1	1	1	5
"Ind_ 378"	1	1	1	1
"Ind_ 379"	1	1	1	1
"Ind_ 380"	1	1	1	5
"Ind_ 381"	1	1	1	1
"Ind_ 382"	1	1	1	1
"Ind_ 383"	1	1	1	5
"Ind_ 384"	1	1	1	5

"Ind_ 385"	1	1	1	1
"Ind_ 386"	1	1	1	1
"Ind_ 387"	1	1	1	5
"Ind_ 388"	1	1	1	5
"Ind_ 389"	1	1	1	1
"Ind_ 390"	1	1	1	1
"Ind_ 391"	1	1	1	5
"Ind_ 392"	1	1	1	5
"Ind_ 393"	1	1	1	5
"Ind_ 394"	1	1	1	1
"Ind_ 395"	1	1	1	5
"Ind_ 396"	1	1	1	5
"Ind_ 397"	1	1	1	5
"Ind_ 398"	1	1	1	5
"Ind_ 399"	1	1	1	1
"Ind_ 400"	1	1	1	1

Fichier "scluster.txt"

400

5

"Ind_ 1"	2	1	3	3
"Ind_ 2"	2	1	3	3
"Ind_ 3"	2	1	3	3
"Ind_ 4"	2	1	3	3
"Ind_ 5"	2	1	3	3
"Ind_ 6"	2	1	3	3
"Ind_ 7"	2	1	3	3
"Ind_ 8"	2	1	3	3
"Ind_ 9"	2	1	3	3
"Ind_ 10"	2	1	3	3
"Ind_ 11"	2	1	3	3
"Ind_ 12"	2	1	3	3
"Ind_ 13"	2	1	3	3
"Ind_ 14"	2	1	3	3
"Ind_ 15"	2	1	3	3
"Ind_ 16"	2	1	3	3
"Ind_ 17"	2	1	3	3
"Ind_ 18"	2	1	3	3
"Ind_ 19"	2	1	3	3
"Ind_ 20"	2	1	3	3
"Ind_ 21"	2	1	3	3
"Ind_ 22"	2	1	3	3
"Ind_ 23"	2	1	3	3
"Ind_ 24"	2	1	3	3
"Ind_ 25"	2	1	3	3
"Ind_ 26"	2	1	3	3
"Ind_ 27"	2	1	3	3
"Ind_ 28"	2	1	3	3
"Ind_ 29"	2	1	3	3
"Ind_ 30"	2	1	3	3
"Ind_ 31"	2	1	3	3
"Ind_ 32"	2	1	3	3
"Ind_ 33"	2	1	3	3
"Ind_ 34"	2	1	3	3
"Ind_ 35"	2	1	3	3
"Ind_ 36"	2	1	3	3
"Ind_ 37"	2	1	3	3
"Ind_ 38"	2	1	3	3
"Ind_ 39"	2	1	3	3
"Ind_ 40"	2	1	3	3
"Ind_ 41"	2	1	3	3
"Ind_ 42"	2	1	3	3
"Ind_ 43"	2	1	3	3
"Ind_ 44"	2	1	3	3
"Ind_ 45"	2	1	3	3
"Ind_ 46"	2	1	3	3

"Ind_ 47"	2	1	3	3
"Ind_ 48"	2	1	3	3
"Ind_ 49"	2	1	3	3
"Ind_ 50"	2	1	3	3
"Ind_ 51"	2	1	3	3
"Ind_ 52"	2	1	3	3
"Ind_ 53"	2	1	3	3
"Ind_ 54"	2	1	3	3
"Ind_ 55"	2	1	3	3
"Ind_ 56"	2	1	3	3
"Ind_ 57"	2	1	3	3
"Ind_ 58"	2	1	3	3
"Ind_ 59"	2	1	3	3
"Ind_ 60"	2	1	3	3
"Ind_ 61"	2	1	3	3
"Ind_ 62"	2	1	3	3
"Ind_ 63"	2	1	3	3
"Ind_ 64"	2	1	3	3
"Ind_ 65"	2	1	3	3
"Ind_ 66"	2	1	3	3
"Ind_ 67"	2	1	3	3
"Ind_ 68"	2	1	3	3
"Ind_ 69"	2	1	3	3
"Ind_ 70"	2	1	3	3
"Ind_ 71"	2	1	3	3
"Ind_ 72"	2	1	3	3
"Ind_ 73"	2	1	3	3
"Ind_ 74"	2	1	3	3
"Ind_ 75"	2	1	3	3
"Ind_ 76"	2	1	3	3
"Ind_ 77"	2	1	3	3
"Ind_ 78"	2	1	3	3
"Ind_ 79"	2	1	3	3
"Ind_ 80"	2	1	3	3
"Ind_ 81"	2	1	3	3
"Ind_ 82"	2	1	3	3
"Ind_ 83"	2	1	3	3
"Ind_ 84"	2	1	3	3
"Ind_ 85"	2	1	3	3
"Ind_ 86"	2	1	3	3
"Ind_ 87"	2	1	3	3
"Ind_ 88"	2	1	3	3
"Ind_ 89"	2	1	3	3
"Ind_ 90"	2	1	3	3
"Ind_ 91"	2	1	3	3
"Ind_ 92"	2	1	3	3
"Ind_ 93"	2	1	3	3
"Ind_ 94"	2	1	3	3
"Ind_ 95"	2	1	3	3
"Ind_ 96"	2	1	3	3
"Ind_ 97"	2	1	3	3
"Ind_ 98"	2	1	3	3
"Ind_ 99"	2	1	3	3
"Ind_ 100"	2	1	3	3
"Ind_ 101"	1	2	1	1
"Ind_ 102"	1	2	1	1
"Ind_ 103"	1	2	1	1
"Ind_ 104"	1	2	1	1
"Ind_ 105"	1	2	1	1
"Ind_ 106"	1	2	1	1
"Ind_ 107"	1	2	1	1
"Ind_ 108"	1	2	1	1
"Ind_ 109"	1	2	1	1
"Ind_ 110"	1	2	1	1
"Ind_ 111"	1	2	1	1
"Ind_ 112"	1	2	1	1
"Ind_ 113"	1	2	1	1

"Ind_ 114"	1	2	1	1
"Ind_ 115"	1	2	1	1
"Ind_ 116"	1	2	1	1
"Ind_ 117"	1	2	1	1
"Ind_ 118"	1	2	1	1
"Ind_ 119"	1	2	1	1
"Ind_ 120"	1	2	1	1
"Ind_ 121"	1	2	1	1
"Ind_ 122"	1	2	1	1
"Ind_ 123"	1	2	1	1
"Ind_ 124"	1	2	1	1
"Ind_ 125"	1	2	1	1
"Ind_ 126"	1	2	1	1
"Ind_ 127"	1	2	1	1
"Ind_ 128"	1	2	1	1
"Ind_ 129"	1	2	1	1
"Ind_ 130"	1	2	1	1
"Ind_ 131"	1	2	1	1
"Ind_ 132"	1	2	1	1
"Ind_ 133"	1	2	1	1
"Ind_ 134"	1	2	1	1
"Ind_ 135"	1	2	1	1
"Ind_ 136"	1	2	1	1
"Ind_ 137"	1	2	1	1
"Ind_ 138"	1	2	1	1
"Ind_ 139"	1	2	1	1
"Ind_ 140"	1	2	1	1
"Ind_ 141"	1	2	1	1
"Ind_ 142"	1	2	1	1
"Ind_ 143"	1	2	1	1
"Ind_ 144"	1	2	1	1
"Ind_ 145"	1	2	1	1
"Ind_ 146"	1	2	1	1
"Ind_ 147"	1	2	1	1
"Ind_ 148"	1	2	1	1
"Ind_ 149"	1	2	1	1
"Ind_ 150"	1	2	1	1
"Ind_ 151"	1	2	1	1
"Ind_ 152"	1	2	1	1
"Ind_ 153"	1	2	1	1
"Ind_ 154"	1	2	1	1
"Ind_ 155"	1	2	1	1
"Ind_ 156"	1	2	1	1
"Ind_ 157"	1	2	1	1
"Ind_ 158"	1	2	1	1
"Ind_ 159"	1	2	1	1
"Ind_ 160"	1	2	1	1
"Ind_ 161"	1	2	1	1
"Ind_ 162"	1	2	1	1
"Ind_ 163"	1	2	1	1
"Ind_ 164"	1	2	1	1
"Ind_ 165"	1	2	1	1
"Ind_ 166"	1	2	1	1
"Ind_ 167"	1	2	1	1
"Ind_ 168"	1	2	1	1
"Ind_ 169"	1	2	1	1
"Ind_ 170"	1	2	1	1
"Ind_ 171"	1	2	1	1
"Ind_ 172"	1	2	1	1
"Ind_ 173"	1	2	1	1
"Ind_ 174"	1	2	1	1
"Ind_ 175"	1	2	1	1
"Ind_ 176"	1	2	1	1
"Ind_ 177"	1	2	1	1
"Ind_ 178"	1	2	1	1
"Ind_ 179"	1	2	1	1
"Ind_ 180"	1	2	1	1

"Ind_ 181"	1	2	1	1
"Ind_ 182"	1	2	1	1
"Ind_ 183"	1	2	1	1
"Ind_ 184"	1	2	1	1
"Ind_ 185"	1	2	1	1
"Ind_ 186"	1	2	1	1
"Ind_ 187"	1	2	1	1
"Ind_ 188"	1	2	1	1
"Ind_ 189"	1	2	1	1
"Ind_ 190"	1	2	1	1
"Ind_ 191"	1	2	1	1
"Ind_ 192"	1	2	1	1
"Ind_ 193"	1	2	1	1
"Ind_ 194"	1	2	1	1
"Ind_ 195"	1	2	1	1
"Ind_ 196"	1	2	1	1
"Ind_ 197"	1	2	1	1
"Ind_ 198"	1	2	1	1
"Ind_ 199"	1	2	1	1
"Ind_ 200"	1	2	1	1
"Ind_ 201"	2	3	4	5
"Ind_ 202"	2	3	4	5
"Ind_ 203"	2	3	4	5
"Ind_ 204"	2	3	4	5
"Ind_ 205"	2	3	4	5
"Ind_ 206"	2	3	4	5
"Ind_ 207"	2	3	4	5
"Ind_ 208"	2	3	4	5
"Ind_ 209"	2	3	4	5
"Ind_ 210"	2	3	4	5
"Ind_ 211"	2	3	4	5
"Ind_ 212"	2	3	4	5
"Ind_ 213"	2	3	4	5
"Ind_ 214"	2	3	4	5
"Ind_ 215"	2	3	4	5
"Ind_ 216"	2	3	4	5
"Ind_ 217"	2	3	4	5
"Ind_ 218"	2	3	4	5
"Ind_ 219"	2	3	4	5
"Ind_ 220"	2	3	4	5
"Ind_ 221"	2	3	4	5
"Ind_ 222"	2	3	4	5
"Ind_ 223"	2	3	4	5
"Ind_ 224"	2	3	4	5
"Ind_ 225"	2	3	4	5
"Ind_ 226"	2	3	4	5
"Ind_ 227"	2	3	4	5
"Ind_ 228"	2	3	4	5
"Ind_ 229"	2	3	4	5
"Ind_ 230"	2	3	4	5
"Ind_ 231"	2	3	4	5
"Ind_ 232"	2	3	4	5
"Ind_ 233"	2	3	4	5
"Ind_ 234"	2	3	4	5
"Ind_ 235"	2	3	4	5
"Ind_ 236"	2	3	4	5
"Ind_ 237"	2	3	4	5
"Ind_ 238"	2	3	4	5
"Ind_ 239"	2	3	4	5
"Ind_ 240"	2	3	4	5
"Ind_ 241"	2	3	4	5
"Ind_ 242"	2	3	4	5
"Ind_ 243"	2	3	4	5
"Ind_ 244"	2	3	4	5
"Ind_ 245"	2	3	4	5
"Ind_ 246"	2	3	4	5
"Ind_ 247"	2	3	4	5

"Ind_ 248"	2	3	4	5
"Ind_ 249"	2	3	4	5
"Ind_ 250"	2	3	4	5
"Ind_ 251"	2	3	4	5
"Ind_ 252"	2	3	4	5
"Ind_ 253"	2	3	4	5
"Ind_ 254"	2	3	4	5
"Ind_ 255"	2	3	4	5
"Ind_ 256"	2	3	4	5
"Ind_ 257"	2	3	4	5
"Ind_ 258"	2	3	4	5
"Ind_ 259"	2	3	4	5
"Ind_ 260"	2	3	4	5
"Ind_ 261"	2	3	4	5
"Ind_ 262"	2	3	4	5
"Ind_ 263"	2	3	4	5
"Ind_ 264"	2	3	4	5
"Ind_ 265"	2	3	4	5
"Ind_ 266"	2	3	4	5
"Ind_ 267"	2	3	4	5
"Ind_ 268"	2	3	4	5
"Ind_ 269"	2	3	4	5
"Ind_ 270"	2	3	4	5
"Ind_ 271"	2	3	4	5
"Ind_ 272"	2	3	4	5
"Ind_ 273"	2	3	4	5
"Ind_ 274"	2	3	4	5
"Ind_ 275"	2	3	4	5
"Ind_ 276"	2	3	4	5
"Ind_ 277"	2	3	4	5
"Ind_ 278"	2	3	4	5
"Ind_ 279"	2	3	4	5
"Ind_ 280"	2	3	4	5
"Ind_ 281"	2	3	4	5
"Ind_ 282"	2	3	4	5
"Ind_ 283"	2	3	4	5
"Ind_ 284"	2	3	4	5
"Ind_ 285"	2	3	4	5
"Ind_ 286"	2	3	4	5
"Ind_ 287"	2	3	4	5
"Ind_ 288"	2	3	4	5
"Ind_ 289"	2	3	4	5
"Ind_ 290"	2	3	4	5
"Ind_ 291"	2	3	4	5
"Ind_ 292"	2	3	4	5
"Ind_ 293"	2	3	4	5
"Ind_ 294"	2	3	4	5
"Ind_ 295"	2	3	4	5
"Ind_ 296"	2	3	4	5
"Ind_ 297"	2	3	4	5
"Ind_ 298"	2	3	4	5
"Ind_ 299"	2	3	4	5
"Ind_ 300"	2	3	4	5
"Ind_ 301"	1	3	2	2
"Ind_ 302"	1	2	2	2
"Ind_ 303"	1	2	2	4
"Ind_ 304"	1	2	2	2
"Ind_ 305"	1	2	2	4
"Ind_ 306"	1	2	2	4
"Ind_ 307"	1	3	2	4
"Ind_ 308"	1	3	2	4
"Ind_ 309"	1	2	2	4
"Ind_ 310"	1	2	2	2
"Ind_ 311"	1	3	2	4
"Ind_ 312"	1	2	2	2
"Ind_ 313"	1	2	2	2
"Ind_ 314"	1	2	2	4

"Ind_ 315"	1	2	2	2
"Ind_ 316"	1	2	2	4
"Ind_ 317"	1	2	2	2
"Ind_ 318"	1	3	2	2
"Ind_ 319"	1	3	2	2
"Ind_ 320"	1	2	2	2
"Ind_ 321"	1	3	2	2
"Ind_ 322"	1	2	2	2
"Ind_ 323"	1	2	2	4
"Ind_ 324"	1	2	2	4
"Ind_ 325"	1	3	2	2
"Ind_ 326"	1	2	2	4
"Ind_ 327"	1	2	2	4
"Ind_ 328"	1	2	2	4
"Ind_ 329"	1	2	2	4
"Ind_ 330"	1	3	2	2
"Ind_ 331"	1	2	2	2
"Ind_ 332"	1	2	2	2
"Ind_ 333"	1	3	2	2
"Ind_ 334"	1	2	2	2
"Ind_ 335"	1	2	2	4
"Ind_ 336"	1	2	2	2
"Ind_ 337"	1	2	2	2
"Ind_ 338"	1	2	2	2
"Ind_ 339"	1	3	2	2
"Ind_ 340"	1	2	2	4
"Ind_ 341"	1	2	2	2
"Ind_ 342"	1	2	2	4
"Ind_ 343"	1	2	2	4
"Ind_ 344"	1	2	2	4
"Ind_ 345"	1	2	2	2
"Ind_ 346"	1	2	2	4
"Ind_ 347"	1	2	2	4
"Ind_ 348"	1	2	2	4
"Ind_ 349"	1	2	2	2
"Ind_ 350"	1	2	2	4
"Ind_ 351"	1	2	2	2
"Ind_ 352"	1	2	2	2
"Ind_ 353"	1	2	2	4
"Ind_ 354"	1	3	2	2
"Ind_ 355"	1	2	2	4
"Ind_ 356"	1	2	2	2
"Ind_ 357"	1	2	2	4
"Ind_ 358"	1	2	2	2
"Ind_ 359"	1	2	2	2
"Ind_ 360"	1	3	2	2
"Ind_ 361"	1	2	2	2
"Ind_ 362"	1	2	2	2
"Ind_ 363"	1	2	2	2
"Ind_ 364"	1	2	2	4
"Ind_ 365"	1	2	2	4
"Ind_ 366"	1	2	2	2
"Ind_ 367"	1	2	2	2
"Ind_ 368"	1	2	2	4
"Ind_ 369"	1	2	2	2
"Ind_ 370"	1	2	2	4
"Ind_ 371"	1	3	2	4
"Ind_ 372"	1	3	2	4
"Ind_ 373"	1	3	2	2
"Ind_ 374"	1	2	2	2
"Ind_ 375"	1	2	2	2
"Ind_ 376"	1	2	2	2
"Ind_ 377"	1	2	2	4
"Ind_ 378"	1	2	2	2
"Ind_ 379"	1	3	2	2
"Ind_ 380"	1	3	2	4
"Ind_ 381"	1	3	2	4

"Ind_ 382"	1	2	2	2
"Ind_ 383"	1	2	2	4
"Ind_ 384"	1	2	2	4
"Ind_ 385"	1	2	2	2
"Ind_ 386"	1	3	2	2
"Ind_ 387"	1	2	2	4
"Ind_ 388"	1	2	2	4
"Ind_ 389"	1	2	2	2
"Ind_ 390"	1	3	2	2
"Ind_ 391"	1	2	2	4
"Ind_ 392"	1	2	2	4
"Ind_ 393"	1	2	2	4
"Ind_ 394"	1	2	2	4
"Ind_ 395"	1	2	2	4
"Ind_ 396"	1	3	2	2
"Ind_ 397"	1	2	2	4
"Ind_ 398"	1	3	2	4
"Ind_ 399"	1	2	2	2
"Ind_ 400"	1	2	2	4

BIBLIOGRAPHIE

- [1] I. Balderjahn, R. Mathar, et M. Schader, eds., *Classification, Data Analysis and Data Highways*, Springer-Verlag, Heidelberg, 1998.
- [2] H.-H. Bock et E. Diday, eds., *Analysis of Symbolic Data*, Springer-Verlag, Heidelberg, 2000.
- [3] H.-H. Bock, Y. Lechevallier, et al., *Report of the meeting ASSO-WP6.2*, tech. rep., Classification Group, Munich, 2001.
- [4] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, et H. Ralambondrainy, *Classification automatique des données*, Dunod (Bordas), Paris, 1989.
- [5] J. Chandon et S. Pinson, *Analyse typologique : théorie et applications*, Masson, Paris, 1995.
- [6] M. Chavent, *Analyse des Données Symboliques : Une méthode divisive de classification*, Thèse de Doctorat, Université de Paris IX-Dauphine, 1997.
- [7] E. Diday, *Nouvelles méthodes et nouveaux concepts en classification automatique et reconnaissance des formes*, Thèse de Doctorat, Université de Paris VI, 1972.
- [8] E. Diday et al., *Optimisation en classification automatique*, vol. 1, Institut National de Recherche en Informatique, Le Chesnay, 1979.
- [9] E. Diday et al., *Optimisation en classification automatique*, vol. 2, Institut National de Recherche en Informatique, Le Chesnay, 1979.

- [10] I. Douny, *Application de la méthode divisive de classification symbolique à des données modales*, Mémoire de fin d'étude, Facultés Universitaires Notre-Dame de la Paix, 1999.
- [11] P. Groenen, H. Kiers, J.-P. Rasson, et M. Schader, eds., *Data Analysis, Classification and Related Methods*, Springer-Verlag, Heidelberg, 2000.
- [12] M. Jambu, *Classification automatique pour l'analyse des données*, vol. 1, Dunod (Bordas), Paris, 1978.
- [13] L. Kaufman et P. Rousseeuw, *Finding Groups in Data*, Wiley-Interscience Publication, New York, 1990.
- [14] V. Mathot, *Une première approche de l'analyse des données symboliques*, Mémoire de fin d'étude, Facultés Universitaires Notre-Dame de la Paix, 1997.
- [15] B. Van Cutsem, ed., *Classification and Dissimilarity Analysis*, Springer-Verlag, New-York, 1994.